

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Strojnícka fakulta

Evidenčné číslo: SjF-13432-104853

**BoBShield: Miniaturizovaný experiment
„gulôčka na tyči“**

Bakalárska práca

2022

Michal Bíró

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Strojnícka fakulta

Evidenčné číslo: SjF-13432-104853

**BoBShield: Miniaturizovaný experiment
„gulôčka na tyči“**

Bakalárska práca

Študijný program: automatizácia a informatizácia strojov a procesov

Študijný odbor: kybernetika

Školiace pracovisko: Ústav automatizácie, merania a aplikovanej informatiky

Vedúci záverečnej práce: Ing. Mgr. Anna Vargová

Konzultant: Ing. Erik Mikuláš

Bratislava 2022

Michal Bíro



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Michal Bíró**
ID študenta: 104853
Študijný program: automatizácia a informatizácia strojov a procesov
Študijný odbor: kybernetika
Vedúca práce: Ing. Mgr. Anna Vargová
Vedúci pracoviska: doc. Ing. Martin Halaj, PhD.
Konzultant: Ing. Erik Mikuláš
Miesto vypracovania: Ústav automatizácie, merania a aplikovanej informatiky (SjF)

Názov práce: **BoBShield: Miniaturizovaný experiment „gulôčka na tyči“**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Úlohou študenta je zrevidovať existujúci hardvér, navrhnúť a realizovať vylepšenia pre prototypizačnú platformu Arduino v rámci open-source projektu „AutomationShield“. Jedná sa o návrh miniaturizovaného laboratórneho experimentu so spätnoväzobným riadením „gulôčka na tyči“ (Ball on beam), spolu s ovládacím softvériom a inštruktážnymi príkladmi.

Študent:

- naštuduje existujúce riešenie a navrhne možné vylepšenia,
- v rámci implementácie vylepšení navrhne plošný spoj v CAD prostredí DipTrace,
- navrhne a otestuje nové mechanické časti zariadenia,
- vytvorí programátorské rozhranie (API) pre Simulink a doplní existujúce riešenie v Arduino IDE.

Rozsah práce: 30 – 50 strán

Termín odovzdania bakalárskej práce: 27. 05. 2022

Dátum schválenia zadania bakalárskej práce: 26. 05. 2022

Zadanie bakalárskej práce schválil: prof. Ing. Cyril Belavý, CSc. – garant študijného programu

Čestné prehlásenie

Vyhlasujem, že predloženú záverečnú prácu som vypracoval samostatne pod vedením vedúceho záverečnej práce, s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú citované v práci a uvedené v zozname použitej literatúry. Ako autor záverečnej práce ďalej prehlasujem, že som v súvislosti s jej vytvorením neporušil autorské práva tretích osôb.

Bratislava, 27. mája 2022

.....
Vlastnoručný podpis

Ďakujem vedúcej mojej bakalárskej práce, Mgr. Ing. Anne Vargovej, za odbornú pomoc pri vypracovaní bakalárskej práce. Chcem podčakovať aj konzultantovi bakalárskej práce, Ing. Erikovi Mikulášovi, za pomoc pri návrhu 3D modelu a zostavovaní hardvéru.

Bratislava, 27. mája 2022

Michal Bíro

Názov práce: Miniaturizovaný experiment "guľôčka na tyči"

Kľúčové slová: - Arduino, ToF, servomotor, guľôčka na tyči, riadenie

Abstrakt: Cieľom bakalárskej práce je vylepšenie existujúceho zariadenia BOBShield, ktoré reprezentuje experiment s názvom „gulôčka na tyči“ a je súčasťou projektu AutomationShield. V prvej kapitole rozoberá samotný experiment a opisuje pôvodné verzie zariadenia. V druhej kapitole hovorí o zmenách, ktoré sme na zariadení vykonali, vysvetľuje dôvody, prečo sme sa pre dané zmeny rozhodli a ukazuje výsledný model nášho zariadenia. V poslednej kapitole sa venuje softvérovej časti zariadenia, kde hovorí o API v programoch Simulink a Arduino IDE, ktoré sme pre zariadenie upravili alebo vytvorili a príkladoch, ktoré sme testovali na novom hardvéri. V závere hovorí o priestore na zlepšenie daného zariadenia v budúcnosti.

Title: Miniature ball on beam experiment

Keywords: - Arduino, ToF, servomotor, ball on beam, control

Abstract: The aim of this bachelor thesis is to improve the existing BOBs-hield device, which represents an experiment called “ball on beam” and is part of the AutomationShield project. In the first chapter analyzes the experiment itself and describes older versions of the device. In next chapter the changes we have made to the device are described. It explains the reasons for their implementation and shows the final model of our device. The last chapter deals with the software of device, which consists of APIs made in programming environments Simulink and Arduino IDE and examples made in these environments. Those were tested on the new hardware. At the end possible improvements that can be made to the device in the future and outcomes of this thesis are summarized.

Obsah

Úvod	1
1 BOBShield	2
1.1 AutomationShield	2
1.2 Experiment - Guľôčka na tyči	3
1.3 Bobshield R2	4
2 R3 hardvér	6
2.1 Schéma zapojenia	6
2.2 Komponenty	8
2.2.1 Servo	10
2.2.2 ToF snímač - VL6180X	12
2.2.3 Gyroskop - MPU 6050	14
2.2.4 Lineárny regulátor napäťia - LDO	15
2.3 Teleso	16
2.4 Prevod rotačného pohybu - servo, trubička	19
2.5 3D prvky	20
2.6 Zoznam komponentov	23
3 BoBshield API	26
3.1 API pre Arduino IDE	26
3.1.1 Filtrácia	26
3.1.2 Knižnica	29
3.2 API pre Simulink	30
3.2.1 Knižnica	31
3.2.2 PID príklad	33
4 Záver	35
Literatúra	36

Úvod

Oblast' automatizácie a automatického riadenia sa rýchlo rozvíja a v modernom svete má veľké zastúpenie vo výrobe. Automatické výrobné linky, prevádzky a závody sú čoraz viac využívané práve kvôli zvyšovaniu produktivity a redukcii vznikajúcich chýb vo výrobe. Aby mohol rozvoj napredovať je potrebné pre danú oblasť podporovať vzdelávanie budúcich technikov a vedcov. Kvalita vzdelávania sa totiž výrazne odrazí v nasledujúcich rokoch ich práce.

Hoci dôležitou časťou vzdelávania je teória riadenia, s ktorou sa študenti univerzít stretávajú na prednáškach, človek si najviac zapamätá, keď si to môže vyskúšať v praxi. Hoci univerzity disponujú vybavením pre praktické ukážky, vstupu do laboratórií pre študentov môže chýbať určitá flexibilita. Z toho dôvodu môže byť náročné sa počas štúdia stretnúť s praxou a to hlavne v časoch, kedy sa študenti nemôžu zúčastniť prezenčnej vzdelávacej formy. Absencia praktických ukážok môže mať za následok neskoršie problémy pri prechode študentov na pracovný trh.

AutomationShield so svojimi zariadeniami prináša jednoduché riešenie tohto problému, keď tvorbou zariadení minimálnych rozmerov a nákladov poskytuje študentom možnosť vyskúšať si riadenie systému v praxi. Vďaka svojej cene nie sú pre školu tieto zariadenia finančne náročné a svojimi šikovnými rozmermi poskytujú študentom možnosť pracovať na nich aj z domova, čo je viac ako vítané pri dištančnej forme štúdia. Momentálne sa v projekte nachádza 11 zariadení a BOBShield je jedným z nich. Zariadenie už prešlo určitým vývojom a momentálne existujú 2 verzie. Hoci pri druhom zariadení prišlo k viacerým zmenám a vylepšeniam oproti prvej verzii, stále poskytuje priestor na vylepšenie. Cieľom práce bolo opäť urobiť analýzu zariadenia a prísť s nápadmi na vylepšenie, či už hardvéru alebo softvéru. V oblasti softvéru išlo o vývoj API v programovacom prostredí Simulink a tvorbu príkladu na riadenie systému. Pri hardvéri sme sa zamerali na použité komponenty, kde sme hľadali ich varianty nachádzajúce sa na trhu a analyzovali sme ako by ich zmena vplývala na chod zariadenia.

Túto tému som si vybral hlavne z dôvodu osobného záujmu o oblasť riadenia. Tiež považujem celú myšlienku AutomationShieldu za veľmi efektívnu a praktickú. Páči sa mi snaha poskytnúť študentom priestor pre rozvoj a vzdelávanie nie len v teoretickej rovine ale aj v praxi. Hoci ide v princípe o jednoduché zariadenia, každé má veľký potenciál a proces vývoja a vylepšovania jednotlivých verzií a prevedení so sebou prináša taktiež drahocenné skúsenosti a vedomosti.

1 BOBShield

BOBShield je zariadenie, ktoré vzniklo ako súčasť vačšieho projektu AutomationShield. Predstavuje nízko nákladový miniaturizovaný systém - Guľôčka na tyči, ktorý slúži ako učebná pomôcka pri predmetoch zaoberejúcich sa riadením systémov[3]. Zariadenie používa servomotor pre otáčanie trubičkou, v ktorej sa pohybuje guľôčka. Cieľom experimentu je nastaviť naklonenie trubičky tak, aby bolo sa guľôčka nachádzala v nami požadovanej pozícii alebo aby sledovala nami určenú referenčnú trajektóriu. Zariadenie svojimi malými rozmermi a nízkymi nákladmi na výrobu poskytuje možnosť pre študentov vyskúšať si riadenie v praxi mimo zariadení v školských laboratóriách, ktorých dostupnosť je výrazne náročnejšia.

1.1 AutomationShield

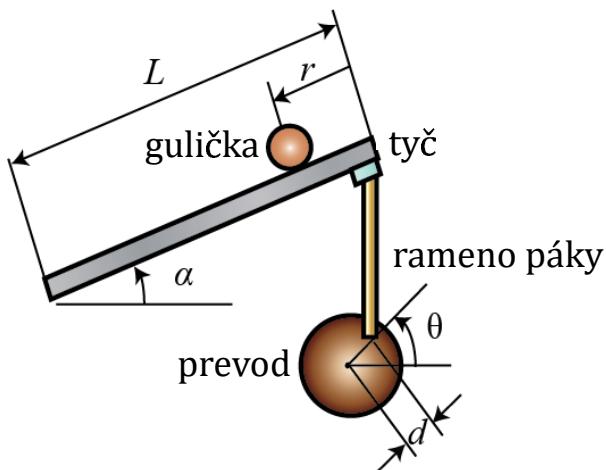
AutomationShield je projekt zameraný na tvorbu jednoduchých a cenovo dostupných zariadení, ktoré slúžia ako učebné pomôcky pri výučbe riadenia systémov. Keďže cieľom je práve dostupnosť pre všetkých, ktorý sa zaujímajú o riadenie a regulovanie systémov, ide o voľne dostupný hardvér a softvér. Základom celého projektu je práve mikropočítač, ktorý na riadenie využívame. Ide o Arduino, taktiež opensource hardvér a softvér, ktorý má medzi študentami pomerne veľkú základňu a je veľmi obľúbenou voľbou pre výučbu mikroprocesorov techniky na univerzitách a školách technického zamerania. Na tvorbu jednotlivých zariadení sa používajú dosky rozširujúce funkciu samotnej Arduino dosky, vo svete všeobecne známe ako „shieldy“. V podstate nejde o nič iné než o dosku plošných spojov (PCB), ktorá sa stará o elektrické spojenie medzi jednotlivými komponentami na zariadení a arduinom. Keďže väčšina typov Arduino dosiek dodržiava zaužívané rozloženie pinov na doske, nie je používanie shieldov obmedzené len je jednu z nich. Z dôvodu dostupnosti je podstatnou časťou tohto projektu práve nízka cena výsledného zariadenia. V priemere dosahujú jednotlivé shieldy cenu okolo 20 eur. Ďalšou časťou sú práve minimálne rozmery zariadení, čo poskytuje jednoduchú manipuláciu a možnosť využívať ich naozaj kdekoľvek, čo samo o sebe robí tento projekt veľmi sympatickým a pre používateľa vytvára dojem jednoduchosti a prístupnosti.

V projekte existuje niekoľko rôznych typov zariadení, ktoré simulujú riadenia rozličných systémov, no či už ide o regulovanie teploty, tlaku alebo pozicie telesa v systéme, jedná sa o miniaturizované verzie väčších systémov. Okrem hardvéru je v AutomationShielde vyvíjaný aj softvér pre riadenie týchto systém-

mov. Medzi programy využívané pre tvorbu API - rozhraní pre programovanie aplikácií patrí v základe Arduino IDE, MATLAB a Simulink.

1.2 Experiment - Guľôčka na tyči

Guľôčka na tyči – po anglicky Ball on Beam, odkiaľ pochádza aj názov nášho zariadenie, je dynamický systém, ktorý pozostáva z dlhej tyče pripojenej k servomotoru, ktorá sa jeho pôsobením nakláňa, a guľôčky, ktorá sa pohybuje po tyči dopredu a dozadu v závislosti od jej naklonenia. Ide o veľmi populárny príklad používaný pri výučbe teórie riadenia. Guľôčka na tyči je teda nestabilný, nelineárny systém s otvorenou slučkou. Aj pri malej výchylke trubičky sa guľôčka skotúľa do jednej z krajných polôh tyče. Stabilizáciu guľôčky v určitej pozícii je potrebný riadiaci systém, ktorý meria polohu guľôčky na tyči a na základe získaných hodnôt vyhodnocuje nastavenie polohy tyče. Na obrázku 1.1 môžeme vidieť parametre systému, kde zmena uhla θ pôsobí cez rameno páky na tyč a spôsobí jej otočenie o uhol α . Keď sa zmení uhol z horizontálnej pozície, gravitácia zapríčiní pohyb guľôčky po tyči. Úlohou riadiaceho systému je teda aby sme vedeli ovládať pozíciu guľôčky.

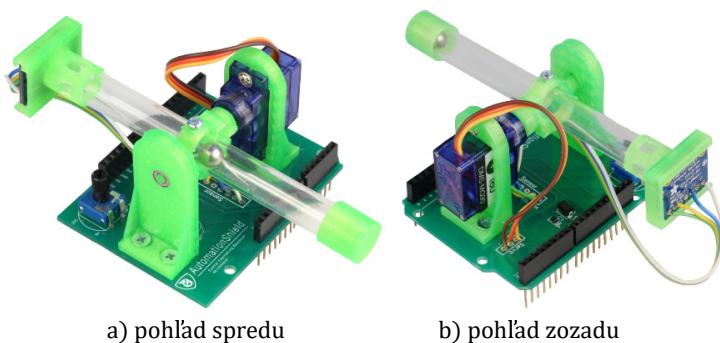


Obr. 1.1: Systém guľôčka na tyči [1]

Náš systém sa od systému na obrázku mierne líši. Namieste tyče sa guľôčka pohybuje uprostred trubičky a jej stred otáčania sa nenachádza na konci trubičky ako to môžeme vidieť na obr. 1.1. Uložená je v držiaku, ktorý sa nachádza v jej strede a práve pomocou neho je na ňu prenášaný krútiaci moment zo servomotoru. V pôvodnej verzii sa teda nenachádzal ani prevod spomenutý na obrázku držiak bol priamo pripojený k servomotoru. Prevod sme však pri vylepšovaní hardvéru zahrnuli do jeho zmien, no oproti klasickému riešeniu uvedenému na obrázku sme zvolil nás vlastný originálny prístup. Výhodou tohto spôsobu zostavenia systému sú jeho rozmery, ktoré sú dostatočne malé aby sme vedeli celý model uložiť na Arduino shield.

1.3 Bobshield R2

Vývoj zariadenia je náročný proces vyžadujúci určitý čas na skúšanie nových možností a vylepšení, či už sa jedná o hardvérovú alebo softvérovú časť. Používaním zariadenia postupne dochádzame k novým nápadom na vylepšenie a tie postupne implementujeme a dolaďujeme, čím sa blížime k ideálnemu riešeniu. Tento stav však takmer nikdy nedosiahneme vzhľadom k neustálemu vývoju a preto je vždy priestor na zmeny a aktualizáciu projektu. Aj BOBShield si už prešiel mnohými zmenami, čo sa použitých komponentov, dizajnu alebo API týka. Prvá verzia BOBShield R1 (obr.1.2) vznikla ako výsledok práce študentov na predmete Mikroprocesorová technika 2, na ústave Automatizácie a merania STU v Bratislave.

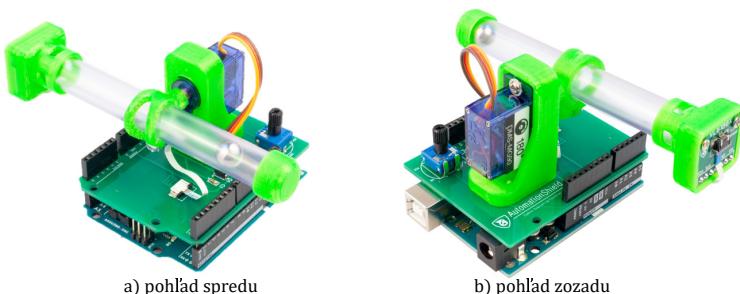


a) pohľad spredu

b) pohľad zozadu

Obr. 1.2: BOBShield R1 [12]

Poslednou verziou je BOBShield R2 (obr. 1.3), ktorá bola vytvorená ako súčasť diplomovej práce na Strojníckej fakulte STU v Bratislave. Oproti prvej verzii prišlo k zmenám hardvéru, kde bolo zmenené rozloženie komponentov na shielde, došlo k vylepšeniu 3D prvkov a zmene veľkosti trubičky a guľôčky. Tiež boli vytvorené nové príklady pre zariadenie v prostredí Arduino IDE ako napríklad identifikácia systému.



a) pohľad spredu

b) pohľad zozadu

Obr. 1.3: BOBShield R2 [12]

Z programových prostredí, ktoré sa v projekte AutomationShield využívajú k riadeniu systému, konkrétnie ide o Arduino IDE, MATLAB a Simulink, bolo vytvorené API (aplikáčné programovacie prostredie) v prostredí Arduino IDE a

knižnica v programe Simulink. Arduino IDE API pozostáva z knižnice BOBS-hield.h a niekoľkých príkladov ako je regulácia alebo identifikácia systému. To ponúka priestor pre tvorbu nových API v už spomenutých programovacích prostrediacach.

Z hľadiska hardvéru ponúka verzia R2 priestor pre analýzu použitých komponentov a ich porovnanie s novými možnosťami ponúkanými na trhu. Tiež je možné zvýšiť presnosť otáčania trubičky o implementáciou už spomínaného prevodu medzi servomotorom a držiakom trubičky. Z toho následne vyplýva aj zmena 3D prvkov a celkového dizajnu zariadenia.

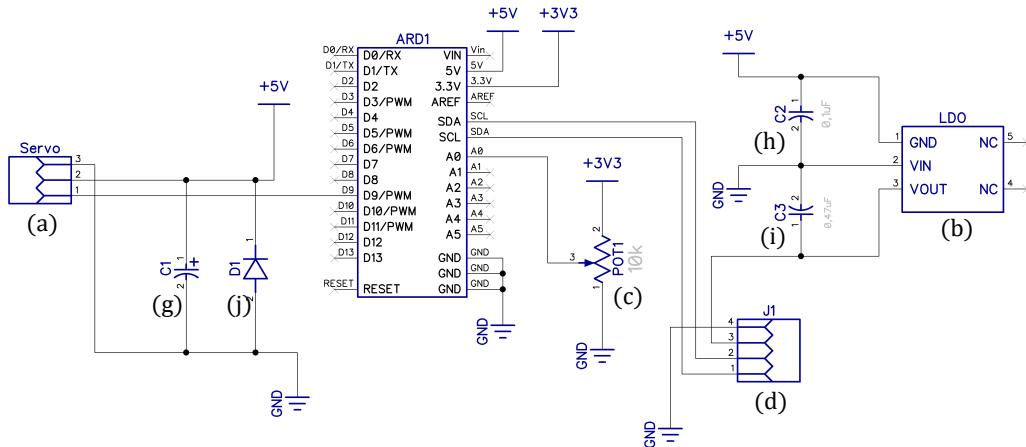
2 R3 hardvér

Nasledujúca kapitola je venovaná vyhotoveniu zariadenia z pohľadu hardvéru. Hovorí o častiach, z ktorých sa zariadenie skladá, o ich parametroch, funkciách a vlastnostiach. Taktiež porovnáva poslednú verziu vyhotovenia – verzia R3, so staršími verziami zariadenia, vysvetľuje, prečo sme sa pre dané zmeny rozhodli a ako vplývajú na celkové fungovanie zariadenia. Medzi výraznejšie zmeny patrí napríklad implementácia nového snímača - gyroskopu a nový typ prevodového systému. Kapitola najprv vysvetľuje schému zapojenia nášho zariadenia, neskôr detailnejšie opisuje jednotlivé komponenty, z ktorých sa zariadenie skladá. Tiež opisuje výber vhodného telesa pre náš systém, implementáciu nového spôsobu prevodu rotačného pohybu zo servomotora na trubičku a tvorbu nových 3D prvkov. V závere sa nachádza zoznam všetkých komponentov a súčiastok, z ktorých sa zariadenie skladá a hovorí o jeho výslednej cene.

2.1 Schéma zapojenia

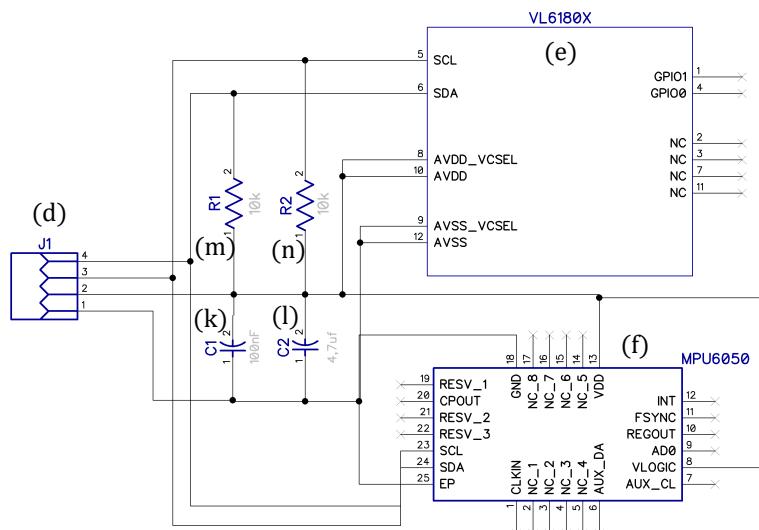
Schéma zapojenia nášho zariadenia prešla od poslednej verzie niekoľkými zmenami. Celkovo by sa dala rozdeliť na 2 oddelené schémy, ktoré sú vzájomne prepojené pomocou FFC (flat flexible cable) kábla. Oddelenie schém je potrebné z dôvodu, že senzory na snímanie polohy guľôčky v trubičke a pootočenia ramena musia byť umiestnené na trubičke. Z toho vyplýva, že schéma celého našeho zariadenia musí byť oddelená, aby mohli byť vytvorené 2 PCB dosky, ktoré sa spájajú do jedného celku. Prvou je časť zariadenia nachádzajúca sa na hlavnej PCB doske (obr. 2.1), ktorú priamo zapájame do Arduina. Obsahuje servomotor (a) , ktorý je napájaný napäťom o veľkosti 5 V a je pripojený k Arduinu prostredníctvom pinu 9, cez ktorý prijíma od Arduina PWM signál, potrebný na riadenie polohy otočenia jeho hriadeľa. K servomotoru je pripojený aj kondenzátor (g) a dióda (j). Úlohou kondenzátora je vyhľadiť prechodové javy, ktoré by mohli ovplyvniť napájanie serva a dióda slúži ako ochrana pred poškodením obvodov spätným EMF – elektromotorickým napäťom. Potenciometer (c) je napájaný napäťom 3,3 V a k Arduinu je pripojený pomocou analógového pinu A0, ktorý slúži na získavanie hodnôt, ktoré neskôr transformujeme do formy, v ktorej môžu byť použité ako referenčná hodnota pre riadenie polohy. Zmenou v schéme zapojenia oproti poslednej verzii je implementácia lineárneho regulátora napäťa – LDO (b), potrebného na zmenu hladiny napäťa pre napájanie snímačov. Je pripojený k zdroju napäťa o veľkosti 5V, ktorého hladinu upravuje a už zmenenú posielá na konektor pre FFC

kábel (d). Ten slúži na prepojenie jednotlivých schém do jedného celku. Je pripojený k pinom SCL a SDA na arduine, ktoré slúžia na sprostredkovanie I2C komunikácie. K LDO sú paralelne pripojené 2 kondenzátory (h) (i), ktoré sú na základe informácií z datasheetu [10] potrebné pre fungovanie LDO a ich parametre sú pre daný lineárny regulátor napäťia zadané.



Obr. 2.1: Schéma hlavnej PCB dosky

Druhá časť schémy (obr. 2.2) je nami navrhnutá breakout doska nachádzajúca sa na konci trubičky, ktorá obsahuje naše snímače. Snímač polohy – VL6180X (e) je pripojený ku konektoru (d), ktorý mu poskytuje napájacie napätie a taktiež sprostredkováva I2C komunikáciu medzi snímačom a Arduinom. Gyroskop – MPU-6050 (f) je rovnako ako snímač polohy napájaný cez konektor (d) a taktiež je paralelne pripojený k 2 kondenzátorom (k) a (l). Ich potreba vyplýva z informácií poskytnutých v datasheet [4], kde je uvedený správny spôsob zapojenia snímača aj parametre kondenzátorov. Rezistory (m) a (n) slúžia ako pull-up rezistory pri I2C komunikácii snímačov s Arduinom.



Obr. 2.2: Schéma PCB dosky so snímačmi

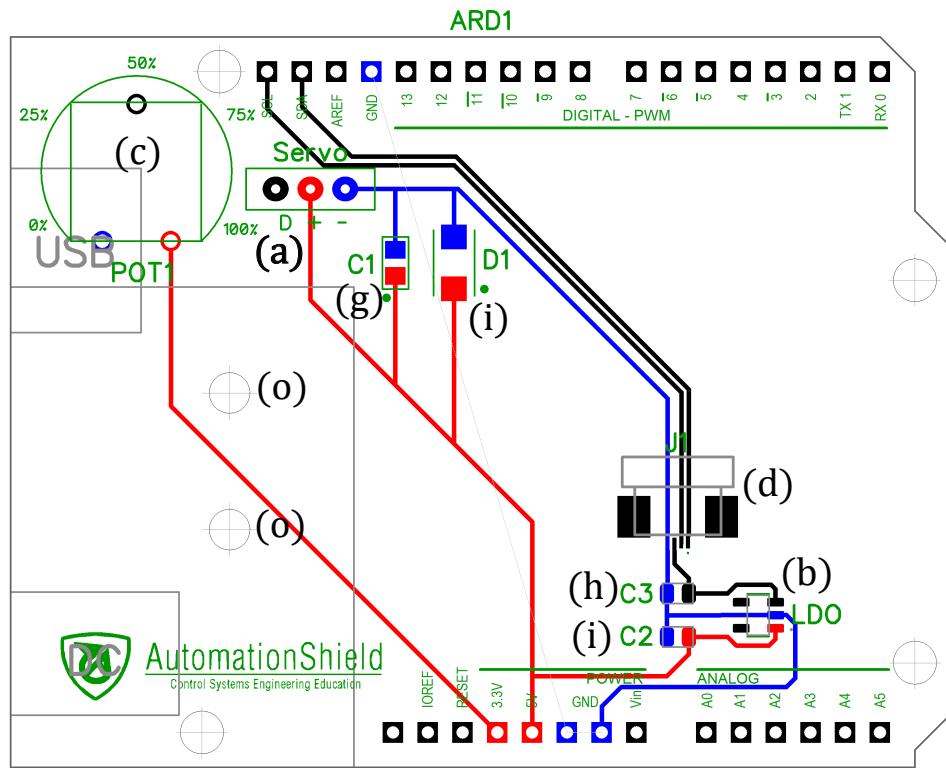
Obe schémy boli navrhnuté vo voľne dostupnom softvéri DIPTrace.

Po návrhu schémy zapojenia jednotlivých dosiek sme pristúpili k ich fyzickému návrhu. Pri tvorbe hlavnej PCB dosky sme museli pracovať s tvarom dosky shieldu pre Arduino, na ktorú sme sa snažili vhodne umiestniť jednotlivé komponenty a konektory. Na hlavnej doske (obr.2.3 a obr.2.4) sa okrem elektronických komponentov budú nachádzať aj 3D prvky zariadenia, preto treba počítať aj s ich umiestnením a vhodne zvoliť diery určené pre montáž týchto prvkov k doske (o). Na doske sa teda nachádza spoj pre pripojenie servomotora (a) a potenciometer (c) pre získavanie referenčnej hodnoty, ktorých umiestnenie sme nechali v rovnakej polohe ako pri pôvodnej verzii dosky. Ich poloha je vzhľadom na umiestnenie 3D prvkov ideálna, keďže konektor servomotora sa nachádza v dostatočnej blízkosti pre jeho pripojenie a zároveň nemusíme použiť zbytočne dlhý kábel. Potenciometer sa tiež vzhľadom k 3D modelu nachádza vo vhodnej pozícii, keďže jeho umiestnenie je ovplyvnené práve potrebou možnosti pohybovať s bežcom počas chodu zariadenia. Preto je potrebné, aby nedochádzalo k vzájomnej interakcii medzi pohyblivými časťami zariadenia a rukou užívateľa pri nastavovaní referenčnej hodnoty. Umiestnenie kondenzátora (g) a diódy (i) je tiež podľa pôvodnej verzie. Umiestnenie konektoru pre FFC káble (d) sme zvolili tak, aby sa po montáži kompletného zariadenia nachádzal pod trubičkou a tak nedochádzalo k jeho zahnutiu a spĺňalo to aj estetickú funkciu. Na základe jeho pozície došlo k umiestneniu spoja pre LDO (b), keďže jeho úlohou je zmena hladiny napäťia do rozsahu napájacích napäťí snímačov, nachádza sa medzi konektorom FFC kábla (d) a zdrojom napäťia z Arduina. Kondenzátory (h) a (i), na základe informácií z datasheetu [10] potrebné pre fungovanie LDO, sú umiestnené v jeho tesnej blízkosti ako je odporúčané.

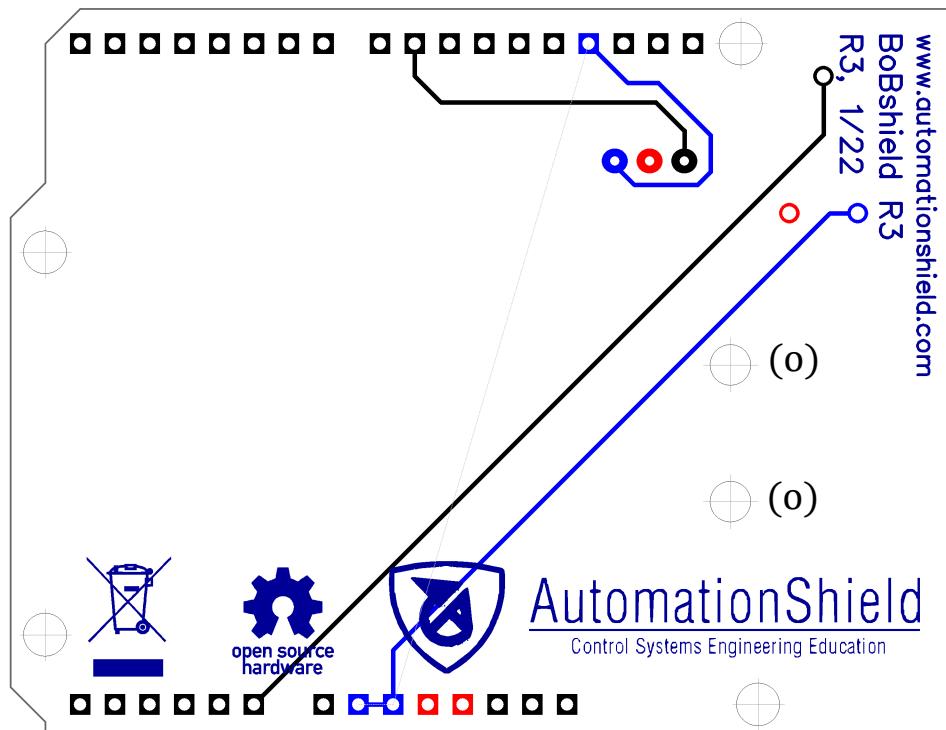
Pri návrhu PCB dosky určenej pre snímače (obr. 2.5), sme neboli obmedzení žiadnym konkrétnym tvarom dosky, preto bolo naším cieľom dosiahnuť rozloženie jednotlivých komponentov optimálnym spôsobom tak aby sme dosiahli čo najmenšie rozmery. Na hornej časti dosky môžeme nájsť plošný spoj pre umiestnenie snímača MPU-6050 (f), konektor pre pripojenie FFC kábla (d) a jednotlivé komponenty (g), (h), (i) a (j), ktorých implementácia vyplýva zo schémy zapojenia jednotlivých snímačov, ktorú je možné nájsť v datasheetoch [4] [9]. Na spodnej strane PCB dosky sa nachádza spoj pre umiestnenie snímača VL6180X (e). Snímač je umiestnený spodnej strane z dôvodu aby smeroval do vnútra trubičky, v ktorej sa nachádza gulička. Po umiestnení PCB dosky do držiaka na konci trubičky, snímač VL6180X smeruje do vnútra trubičky a naopak FFC kábel smeruje von z trubičky, čo poskytuje jeho jednoduchú montáž.

2.2 Komponenty

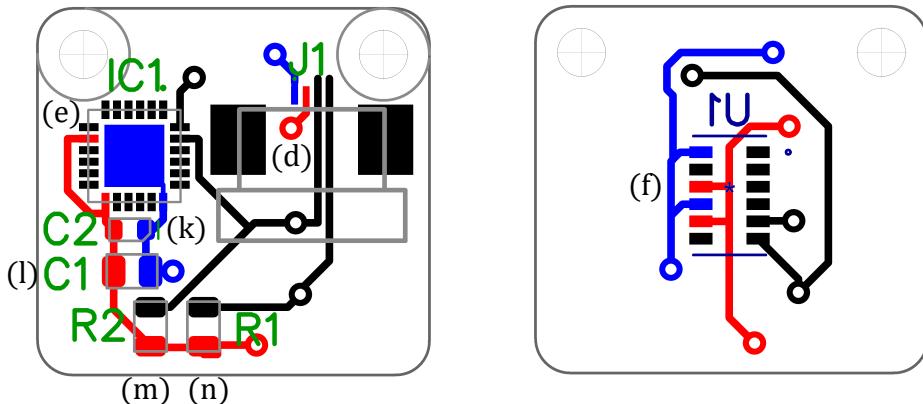
Nasledujúca podkapitola detailnejšie opisuje vlastnosti komponentov, s ktorými sme sa už stretli pri schéme zapojenia. Hovorí o ich parametroch, funkciách, ktoré v zariadení zastrešujú a tiež opisuje proces ich výberu a dôvod



Obr. 2.3: PCB doska - pohľad zhora



Obr. 2.4: PCB doska - pohľad zdola



Obr. 2.5: PCB doska pre snímače

prečo sme si pre naše zariadenie zvolili práve tieto komponenty.

Pre riadenie a regulovanie akéhokoľvek systému je potrebné mať vstup do systému, ktorým nastavujeme zásah do systému. O túto časť riadiacej slučky sa starajú aktuátory. Výstup zo systému zase prezentujú hodnoty, ktoré získavame pomocou snímačov a na ich základe môžeme vyhodnocovať aktuálny stav systému.

V našom zariadení sa o jednotlivé funkcie starajú nasledujúce komponenty:

- snímače – Tof (time of flight) snímač vzdialenosťi, Gyroskop
- aktuátory – Servomotor

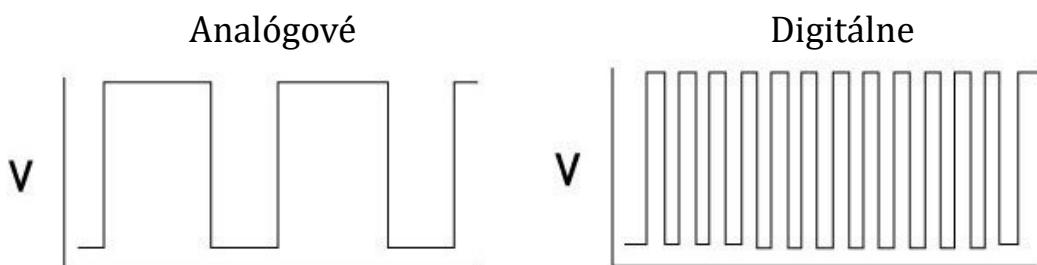
2.2.1 Servo

Servomotor patrí spolu so snímačmi ku komponentom, ktorých vlastnosti výrazne vplývajú na kvalitu riadenia systému. V našom systéme plní funkciu aktuátora, teda sa stará o akčný zásah do systému a tak ovplyvňuje náklon trubičky a následne pohyb guľôčky, ktorej polohu riadime. Na základe dôležitosti tohto komponentu sme sa rozhodli pre analýzu ponúkaných možností na trhu a ich následné porovnanie so servomotorom v pôvodnej verzii. Vo verzii zariadenia R2 bol použitý analógový servomotor s kovovými prevodmi MG-90. Ide o veľmi rozšírený servomotor bežne sa nachádzajúci v začiatočníckych arduino kitoch, určených pre domáce projekty. Z parametrov (tab. 2.1) môžeme vidieť, že splňa požiadavku malých rozmerov, taktiež podľa rozsahu pracovného napäťia odpadá potreba jeho úpravy a servo môže byť napájané priamo z Arduina. Vďaka kovovým prevodom je jeho životnosť vyššia a dosahuje aj lepšie presnosť v porovnaní s plastovými verziami. Čím však servo najviac vyniká oproti iným typom a verziám je jeho nízka cena.

Zmena, ktorú sme zvažovali bola voľba digitálneho servomotora namiesto analógového. Ide o dilemu, s ktorou sa pri výbere servomotora stretáme veľmi často. Každý typ má svoje výhody a nevýhody no najväčším rozdielovým parametrom, od ktorého závisí výber typu serva je frekvencia PWM signálu, s

Hmotnosť [g]	12
Rozmery [mm]	22.3 x 11.8 x 26.3
Krútiaci moment [kg/cm]	1,8 - 2,2
Rýchlosť [s/60 °]	0,12 - 0,8
Prevádzkové napätie [V]	4,8 - 6,0
Typ	analógové
Prevody	kovové
Cena [euro]	6

Tabuľka 2.1: Parametre - MG-90 [2]



Obr. 2.6: Dĺžka cyklu PWM signálu pre analógové a digitálne servo [5]

ktorým dané servo pracuje. S tým súvisia aj ostatné výhody a nevýhody, no podstata je ukrytá práve za týmto faktorom.

Princíp fungovania nášho servomotora je založený na prijímaní napäťového signálu, ktorý prichádza v podobe PWM (pulse width modulation – impulzová šírková modulácia) signálu. PWM signál posielá motoru elektrický impulz rôznej šírky, na základe ktorej sa otočí hriadeľ do polohy, ktorú požadujeme [5]. Analógové servomotory pracujú so signálom o frekvencii 50 Hz, čo znamená že periódna jedného cyklu je 0,02 s, čo sa môže zdať ako dostatočne malý čas, no pri riadení systému, tak dynamického ako je ten náš, ide o pomerne veľký časový interval. Pri digitálnych servách dosahuje PWM signál výrazne vyššie frekvencie – pohybujeme sa v stovkách hertzov. Ak si zoberieme PWM signál o frekvencii 500 Hz, periódna jedného cyklu je 0,002 s, čo je 10 krát viac cyklov za rovnaký čas ako pri analógovom servomotore. To má za následok zníženie času kedy do serva nie je posielaný žiadny elektrický signál. Graficky znázornený rozdiel medzi dĺžkami cyklov PWM signálu pre digitálne a analógové servomotory môžeme vidieť na obrázku 2.1. Zvýšením frekvencii teda digitálne servá dosahujú oveľa rýchlejšie reakcie na zmeny požadovanej polohy, lepšie rozlíšenie a väčší krútiaci moment. Na druhej strane však s väčšou frekvenciou prichádza aj väčšia spotreba energie, vyššia cena a hluk pri prevádzke.

My sme sa rozhodli pre zmenu z analógového na digitálne servo z dôvodu potreby zrýchlenia reakcie aktuátora na zmenu polohy guľôčky, lepšieho roz-

Hmotnosť [g]	13,6
Rozmery [mm]	23.8 x 12 x 25.4
Krútiaci moment [kg/cm]	1 - 5
Rýchlosť [s/60 °]	0,05 - 0,10
Prevádzkové napätie [V]	4,8 - 6,0
Typ	digitálne
Prevody	kovové
Cena [euro]	16

Tabuľka 2.2: Parametre - SH-0253 [8]

líšenia pulzu a väčší krútiaci moment. Za servomotor sme si preto zvolili SH-0253, digitálne servo značky SAVOX. Jeho parametre môžeme vidieť uvedené v tabuľke 2.2. Pri porovnaní parametrov jednotlivých servomotorov môžeme vidieť výrazný rozdiel v rýchlosťach a krútiacich momentoch. Digitálne servo dosahuje podstatne vyššie rýchlosťi otáčania aj krútiaci moment, ktorý je viac ako dvojnásobne väčší.

2.2.2 ToF snímač - VL6180X

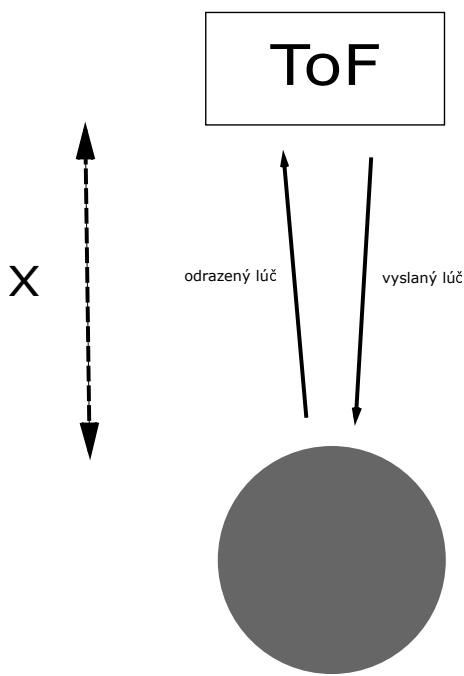
Úlohou snímača je snímať polohu guličky v trubičke a dátu posielat mikropočítaču. Poskytuje nám aktuálnu hodnotu vzdialenosťi, ktorá sa porovnáva s požadovanou hodnotou na základe čoho regulátor vypočíta hodnotu akčného zásahu do systému.

Pri snímači vzdialenosťi sme sa rozhodli ostať pri pôvodnej voľbe typu a modelu, ktorým je VL6180X (obr. 2.8). Ide o ToF snímač, ktorý meria vzdialenosť na základe vysielania, odrazu a následného prijímania svetelného lúča. Princíp fungovania (obr. 2.7) je založený na meraní času, za ktorý svetelný lúč vyslaný zo snímača doletí k telesu, ktorého pozíciu meriame, odrazí sa od neho a vráti sa späť. Na základe rovnice (rov. 2.1), kde c predstavuje rýchlosť svetla, ΔT čas, ktorý trvá lúču odraz a návrat k snímaču, samotný snímač vypočíta vzdialenosť x , v ktorej sa nachádza teleso od snímača. Komunikácia s arduinom prebieha prostredníctvom I2C protokolu. Rozsah napájacieho napäťia je 2,6V – 3,0V.

$$2x = c \cdot \Delta T \quad (2.1)$$

Hoci sme zvažovali aj iné typy snímačov vzdialenosťi ako napríklad ultra-zvukové alebo IR (infračervené) snímače, prišli sme k záveru, že najvhodnejším typom snímača bude práve ToF vďaka jeho malým rozmerom. Z možností ToF snímačov ponúkaných na trhu je pre naše zadanie VL6180X momentálne ideálnou možnosťou z dôvodu:

- merateľného rozsahu - 100 mm,
- presnosti merania - na 1 mm,



Obr. 2.7: Princíp fungovania ToF senzora

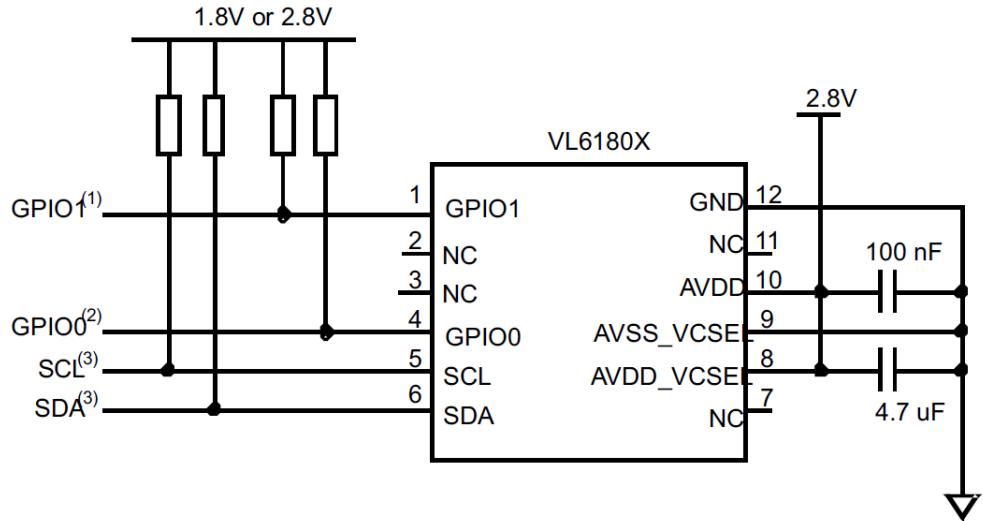


Obr. 2.8: ToF snímač - VL6180X[9]

- malým rozmerom – 4.8 x 2.8 x 1.0 mm [9]

Merateľný rozsah snímača je pre nás úplne postačujúci, keďže guľôčka sa pohybuje v trubičke, ktorá má dĺžku práve 100 mm s tým že v koncových bodoch sa nachádzajú ešte uzavretia trubičky, takže rozsah pohybu guličky je približne 90 mm, čo nám dáva ešte určitú rezervu. Veľa snímačov nespĺňa naše požiadavky práve z dôvodu ich merateľného rozsahu, ktorý nezačína v 0 mm ale až pri vyšších hodnotách. Hoci presnosť merania na 1 mm nie je pre riadenie systému ideálna, z ponúkaných senzorov sa nám nepodarilo v danej kategórii nájsť žiadny s lepšou presnosťou. Rozmery sú pre nás dôležité z hľadiska potreby umiestnenia snímača do jednej z koncových polôh trubičky, aby sme mohli merať pozíciu guľôčky. Taktiež príliš veľké rozmery, ako napríklad pri ultrazvukovom snímači, by pre minimálne rozmery nášho zariadenia neboli vhodné z hľadiska estetickosti. Vďaka minimálnym rozmerom snímača sa nám oproti pôvodnej verzii podarilo zmeniť diel upevnenia snímača na trubičke, čím sme zlepšili aj estetickú zložku zariadenia.

Na obrázku 2.9 môžeme vidieť schému zapojenia snímača uvedenú v datas-



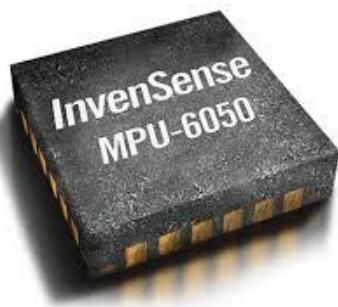
Obr. 2.9: Schéma zapojenia snímača [9]

heete. V našej aplikácii sme piny 1 a 4 (GPIO1 a GPIO0) nepoužívali a nechali sme ich voľné. Piny 5 a 6 – SCL a SDA, slúžiace na I2C komunikáciu snímača s mikropočítačom sú pomocou pull up rezistorov pripojené k napätiu. Za hodnoty pull up rezistorov sme zvolili odpór $10\text{ k}\Omega$. Ako môžeme vidieť v schéme piny 9 a 12 sú uzemnené a piny 10 a 8 sú zase pripojené k napájaniu, pričom sú k nim paralelne pripojené 2 kapacitóry, ktorých hodnota je uvedená pre daný snímač v datasheete. Taktiež je uvedené, aby ich pozícia bola čo možno najbližšie k daným pinom.

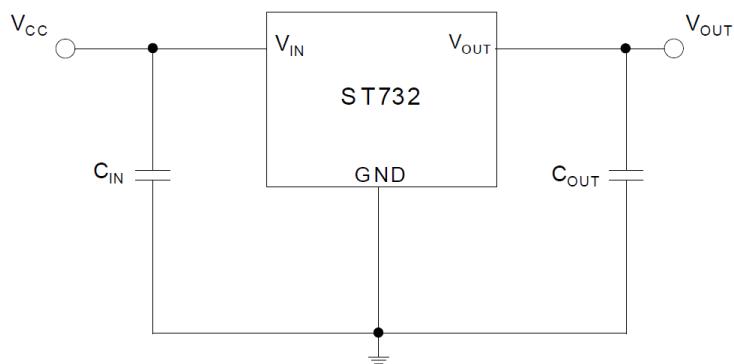
2.2.3 Gyroskop - MPU 6050

Snímač MPU 6050 (obr. 2.10) je oproti pôvodným verziám zariadenia úplnou novinkou. Ide o 6 osový snímač pohybu – 3 osový gyroskop a 3 osový akcelerometer zostavený v malom balení o rozmeroch $4 \times 4 \times 0,9$ mm. Jeho využitie je veľmi široké a stretnúť sa s ním môžeme pri smartfónoch alebo tabletcoch. Poskytuje funkcie využívané pri aplikáciách ako navigácia, rozšírená realita, monitorovanie zdravia a pohybu a mnoho ďalších. Snímač užívateľovi poskytuje možnosť nastavenia meraného rozsahu v závislosti od aplikácie, pre ktorú bude využitý, teda pre sledovanie jednak rýchlych aj pomalých pohybov. Pri gyroskope si užívateľ môže vybrať z nasledujúcich rozsahov - ± 250 , ± 500 , ± 1000 a $\pm 2000^\circ/\text{s}$. U akcelerometra ide o rozsahy $\pm 2\text{g}$, $\pm 4\text{g}$, $\pm 8\text{g}$ a $\pm 16\text{g}$. Komunikácia s arduinom prebieha rovnako ako pri ToF snímači prostredníctvom I2C protokolu a rozsah napájacieho napätiia $2,375\text{ V} - 3,46\text{ V}$ [4].

Jeho úlohou je merať uhol natočenia trubičky, na základe čoho vieme nastaviť trubičku do vodorovnej polohy.



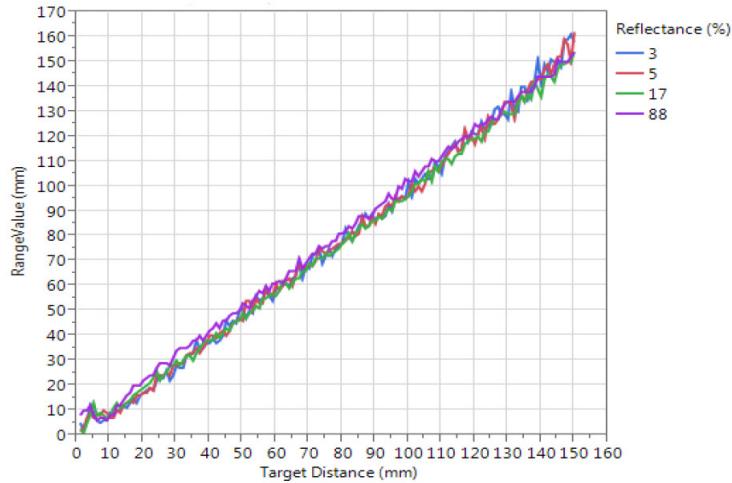
Obr. 2.10: MPU 6050 [7]



Obr. 2.11: Schéma zapojenia LDO [10]

2.2.4 Lineárny regulátor napäťia - LDO

Z dôvodu tvorby vlastného braekoutu, na ktorom sú umiestnené oba naše snímače, bolo potrebné nájsť vhodné napájacie napätie tak, aby vyhovovalo obom snímačom a spadalo do ich rozsahov napäti. Pre snímač VL6180X je vstupné napätie v rozmedzí 2,6 V - 3,0 V [9] a pri MPU6050 ide o rozsah 2,375 – 3,46 V[4]. Arduino nám ponúka len 2 úrovne napájania a to 3,3 V a 5 V. Ak by sme využívali len snímač MPU6050 mohli by sme priamo využiť napájanie 3,3 V, ktoré patrí do jeho pracovného rozsahu, no pre snímač polohy je táto možnosť nevyhovujúca. Preto je potrebné upraviť ponúkané úrovne napäťia z arduino dosky do rozsahu 2,6 - 3,0 V, vyhovujúcemu obom snímačom. O zmenu napäťia sa stará lineárny regulátor napäťia – STM732M28R od firmy STMicroelectronics , ktorý vstupné napätie v rozsahu 2,5 – 28 V prevádzza na úroveň 2,8 V. Táto hladina je vyhovujúca pre oba snímače. Spôsob zapojenia (obr. 2.11) sme získali z datasheetu [10], ktorý nám uvádzza správne zapojenia regulátora a komponenty potrebné pri jeho implementácii. Ide o 2 kondenzátory, ktorých parametre a umiestnenie v schéme sú uvedené v datasheete [10].



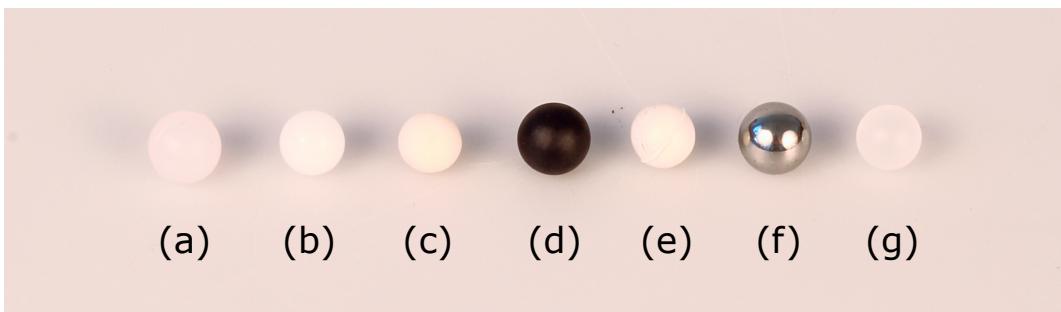
Obr. 2.12: Závislosť merania od odrazivosti telesa[9]

2.3 Teleso

Dôležitou časťou celého systému je práve gulička, pohybujúca sa v trubičke, ktorej polohu sledujeme a riadime. Na náš systém vplýva jej hmotnosť a tvar ale aj kvalita a farba jej povrchu. Keďže na meranie polohy guľôčky používame ToF snímač, ktorý meria vzdialenosť na základe času, za ktorý sa svetelný lúč vyslaný zo senzora odrazí od telesa a vráti naspäť, na kvalitu merania vplývajú aj tieto parametre.

Výrobca v datasheete [9] uvádza nezávislosť merania od farby alebo kvality povrchu telesa. Deklarované je to aj grafom na obrázku 2.12 , kde môžeme vidieť, že nie je signifikantný rozdiel medzi meraniami pri rôznych odrazivostach povrchu.

Aj napriek tomu sme v našich meraniach mohli sledovať odlišnosti v presnosti pre rôzne typy guľôčok. Tento fakt môže byť ovplyvnený práve tvarom meraného telesa, ktorý pre funkčnosť snímača nie je ideálny, no pre potreby nášho zadania nevyhnutný. Svetelný lúč nedopadá na kolmý povrch, preto nemusí byť odrazený práve pod takým uhlom, aby ho dokázal snímač adekvátne zachytiť a zanalyzovať. Na základe tohto faktu sme predpokladali, že by ideálnym riešením bolo teleso s povrhom, ktorý čo najviac rozptýli dopadajúci lúč aby šanca, že sa lúč od neho odrazí k snímaču, bola čo najvyššia. Tým by sa zlepšila kvalita merania. Ďalšou požiadavkou pri hľadaní ideálneho telesa bola dostatočná hmotnosť guľôčky tak, aby na zmenu stavu systému reagovala v reakčnom čase dostatočne dlhom, aby náš riadiaci systém stíhal vykonávať potrebné výpočty. Zároveň nechceme aby bol zbytočne dlhý. Na túto požiadavku vplýva aj kvalita povrchu guľôčky. Aby sa guľôčka mohla voľne pohybovať v trubičke je potrebné, aby bol jej tvar čo najviac podobný tvaru ideálnej gule. Poslednou požiadavkou bola jeho jednoduchá dostupnosť. Keďže sa jedná o open source projekt, je potrebné aby sa každý, kto by mal záujem o zstrojenie nášho zariadenia, vedel k danej guľôčke bez väčších problémov dostať.



Obr. 2.13: Guličky

Z možností dostupných na trhu sme sa rozhodli otestovať náš snímač pre viacero typov materiálov (obr.2.13): silikón (a), POM - polyoxymetylén (b), drevo (c), NBR - butadien-akrilonitrilový kaučuk (d) (syntetická guma), NR - prírodná guma (e), oceľ (f) a PP - polypropylén (g).

Pri meraní sme postupovali nasledovne. Guľôčku sme nastavili do krajnej polohy v trubičke (vzdialenosť 90 mm), následne sme pomocou ToF snímača zmerali 100 hodnôt jej aktuálnej polohy. Hodnoty sme vložili do tabuľky a pomocou vzorca 2.2 sme vypočítali smerodajnú odchýlku pre dané meranie.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (2.2)$$

Tento postup sme zopakovali pre všetky materiály. V tabuľke 2.3 môžeme vidieť porovnanie smerodajných odchýlok pre jednotlivé materiály. Na základe merania, ktoré sme vykonali, môžeme tvrdiť, že náš snímač dosahuje najmenšie smerodajné odchýlky pri materiály PP – polypropylén a najväčšie odchýlky pri NBR – syntetická guma.

Materiál	smerodajná odchýlka
silikón	1,27
POM	1,2891
drevo	1,3644
NBR	1,6271
NR	1,3322
oceľ	1,5
PP	1,2595

Tabuľka 2.3: Smerodajné odchýlky pre dané materiály

Pri výbere guľôčok sme hľadeli aj na priemernú rýchlosť ich pohybu. Ak je guľôčka príliš rýchla, riadiaci systém má menej času reagovať na zmene jej polohy, ktorá sa hlavne pri väčšej zmene referenčnej hodnoty výrazne mení. Keďže dĺžka trubičky je pomerne malá (100 mm), ak riadiaci systém nedokáže včas zareagovať na zmene jej polohy, bude dochádzať k nárazom guľôčky do krajných polôh trubičky, čo je pri riadení nežiaduci jav. Z toho dôvodu sa

snažíme voliť guľôčku, ktorej rýchlosť je adekvátna pre našu aplikáciu. Rýchlosť guličky je závislá jednak od jej hmotnosti, materiálu, kvality povrchu a rozmerov.

Pri meraní sme zaznamenávali polohu guľôčky a čas, v ktorom sa v danej polohe nachádzala. Guľôčka sa pohybovala z jednej krajnej polohy trubičky do druhej krajnej polohy pri naklonení trubičky o uhol 5° . Následne sme namerané hodnoty vykreslili do grafu v programe MATLAB a pomocou jeho analýzy sme našli čas, za ktorý guľôčka prešla danú vzdialenosť trubičky. Postupovali sme nasledovne: V grafe sme si našli body kedy sa guľôčka nachádzala v vzdialosti 80 mm a 20 mm od snímača. Z grafu sme si vypočítali za aký čas guľôčka túto vzdialosť 60 mm prešla. Na základe získaných hodnôt sme vypočítali rýchlosť jednotlivých guľôčok 2.4.

Materiál	primerná rýchlosť [m/s]
silikón	0.1348
POM	0.1504
drevo	0.1412
NBR	0.0917
NR	0.1279
ocel	0.1348
PP	0.1412

Tabuľka 2.4: Rýchlosť guličiek pre dané materiály

V tabuľke 2.4 môžeme vidieť porovnanie rýchlosť jednotlivých guľôčiek pri pohybe mierne naklonenou trubičkou. Najpomalšie sa pohybujúcou je guľôčka z NBR, ktorá oproti guľôčke z ocele, použitej v pôvodných verziách zariadenia dosahuje výrazne nižšie rýchlosť pohybu v trubičke.

Na základe vykonaných meraní sme sa rozhodli zvoliť si za teleso v systéme guľôčku vyrobenú z polypropylénu (PP). Snímač dosahuje pri danej guľôčke najmenšie smerodajné odchylky, takže meraná poloha guľôčky sa najviac približuje jej skutočnej polohe. Meranie s čo najmenšou smerodajnou odchylkou je pri riadení systému kľúčové, no treba uvažovať aj dynamiku guľôčky. Z nameraných rýchlosťí guľôčiek by pre nás bola najvhodnejšou voľbou guľôčka z NBR, no jej presnosť merania sú výrazne horšie v porovnaní s guľôčkou PP. Hoci na základe meraní nedosahuje najlepšie rýchlosť pre našu aplikáciu, rozhodli sme sa zvoliť si ju na základe dobrých výsledkov pri smerodajnej odchylke merania. Dobrou voľbou by mohli byť aj guľôčky zo silikónu alebo materiálu NR. Tie však nespĺňajú požiadavky na tvar guľôčky, keďže pri ich výrobe na ich povrchu ostáva stopa po forme, ktorá následne bráni plynulému pohybu guľôčky v trubičke.

Ak porovnáme guľôčku z ocele, použitú v pôvodnej verzii, a guľôčky z PP, ktorú sme na základe meraní vybrali, môžeme pozorovať výrazné zlepšenie merania snímača z hľadiska smerodajných odchýlok.

2.4 Prevod rotačného pohybu - servo, trubička

Aktuátorom v našom zariadení je servo motor. Aby mohol vytvárať akčný zásah (pootočenie trubičky) do systému, ktorým je naša trubička s guľôčkou, musí prísť ku prenosu rotačnému pohybu.

V pôvodnej verzii zariadenia bol tento prevod riešený priamym upevnením trubičky na servomotor. To znamená, že otočenie trubičky sa rovnalo otočeniu serva o príslušný uhol z rozsahu 30° , ktorý bol stanovený. Najmenší možný uhol pootočenia trubičky sa teda rovnal presnosti serva : 1° . Vzhľadom na malé rozmery trubičky ide o pomerne veľký uhol. Malé možné rozlíšenie uhlia, o ktorý vieme otočiť trubičku, so sebou prináša obmedzenia pri riadení systému, kedy akčný zásah nemôže nadobúdať takú množinu hodnôt. To má za následok neschopnosť nastaviť tak presný akčný zásah, aby sa čo najviac približoval hodnote vypočítanej riadiacim systémom.

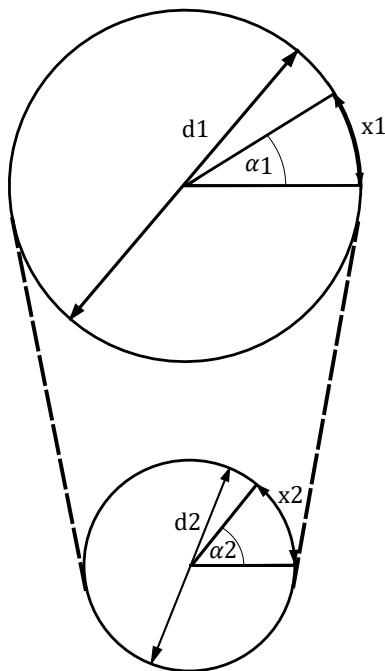
Našim riešením tohto problému je použitie jednoduchého prevodu, cez ktorý budeme prenášať rotačný pohyb serva na trubičky. Prevod sa skladá z 2 koliesok rozdielnych priemerov, pričom jedno je upevnené na servo motor a k druhému je pripojená trubička. Prenos pohybu z jedného kolieska na druhé je riešený pomocou ozubeného remeňa GT2 o šírke 6 mm, bežne používanom pri krokových motoroch alebo 3D tlačiarňach. Výhodou nášho riešenia je využitie takmer celého rozsahu pohybu nášho servo motora. Kým pri pôvodnom zariadení išlo o využitie rozsahu 30° z možných 90° , v danom riešení využívame rozsah 75° .

Prevodový pomer - p a následne aj rozmery koliesok (d_1 - vstupný priemer, d_2 - výstupný priemer) sme vypočítali na základe vzorcov (2.3) pre výpočet prevodov, kde najprv zistíme prevodový pomer na základe vstupného a výstupného rozsahu otočenia. Vzorce sa dajú jednoducho odvodiť na základe dĺžky posunutia remeňa - x , ktorá je na oboch kolieskach vždy rovnaká. Ide o dĺžku kružnicového oblúka, o priemere daného kolieska, ktorého uhol je rovný uhlu otočenia (obr.2.14). Tento uhol môžeme pri výpočtoch nahradiať rozsahom otáčania jednotlivých koliesok. Za výstupný rozsah sme si zvolili 15° . Pre dĺžku trubičky je tento uhol postačujúci, keďže referenčná hodnota sa pohybuje od 0 po 100 mm a guľôčka sa pohybuje v trubičke dostatočne veľkou rýchlosťou bez väčšieho odporu. Nemusí dochádzať k tak výrazným zásahom aktuátora ako je naklonenie o celých 30° , či už kladným alebo záporným smerom. Reakcia systému je aj pri nami zvolenom rozsahu dostatočne rýchla. Pre vstupný rozsah sme si zadali 75° aby sme disponovali určitou rezervou voči maximálnemu rozsahu otočenia.

$$p = \frac{d_1}{d_2} \quad (2.3)$$

$$x_1 = \frac{\pi \cdot d_1 \cdot \alpha_1}{360} \quad (2.4)$$

$$x_2 = \frac{\pi \cdot d_2 \cdot \alpha_2}{360} \quad (2.5)$$



Obr. 2.14: Posunutie remeňa pri rotácii kolieska

$$x_1 = x_2 \quad (2.6)$$

$$p = \frac{d_1}{d_2} = \frac{\alpha_1}{\alpha_2} \quad (2.7)$$

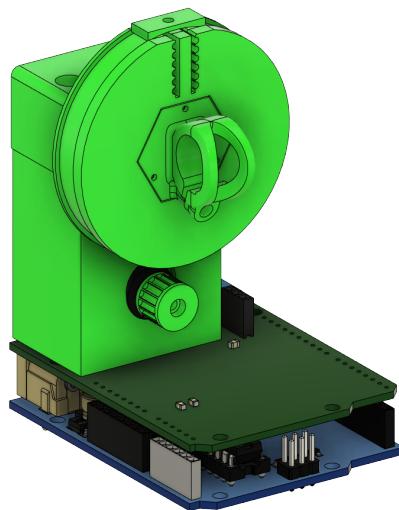
$$p = \frac{30^\circ}{150^\circ} = 0.2 \quad (2.8)$$

Na základe prevodového pomeru vieme povedať, že na 1° otočenia serva prislúcha $0,2^\circ$ otočenia trubičky. Týmto prevodom sme teda 5 násobne zväčsili množinu hodnôt, ktoré môže aktuátor nadobudnúť, a tým zvýšili presnosť akčného zásahu do systému. Ak poznáme prevodový pomer, rozmerky koliesok získame voľbou priemeru jedného z nich a následným výpočtom priemeru druhého kolieska na základe vzorca 2.6. Za priemer kolieska pripojeného k servo motoru sme si zvolili $d_1 = 10mm$. Je to dostatočná veľkosť pre uchytenie kolieska a prevod pohybu pomocou remeňa, no zároveň šetríme miesto aby sme zachovali minimálne rozmerky pôvodnej verzie zariadenia. Z toho vyplýva, že priemer druhého kolieska bude $d_2 = 50mm$.

2.5 3D prvky

Z dôvodu zmeny hardvéru, ako je napríklad vytvorenie vlastnej PCB dosky pre snímače, došlo k potrebe návrhu nového 3D modelu zariadenia a aktualizácií jeho starších prvkov. Najvýraznejšou zmenou je použitie prevodu (kap. 2.4) na prenos rotačného pohybu zo serva na trubičku. Pre implementáciu tohto prevodu bolo potrebné vytvoriť hned' niekoľko nových 3D prvkov a niektoré

staršie prešli menšími zmenami. Tiež sme pri ich návrhu hľadeli na viacero požiadaviek kladených na tento prevodový systém, ako je potreba možnosti napínania remeňa alebo jeho upevnenie na prevodovom koliesku. Na obrázku 2.15 môžeme vidieť už výslednú formu 3D modelu zariadenia, ktorá sa skladá z niekoľkých samostatných prvkov, ktorých funkcie si vysvetlíme v tejto sekcií. Všetky prvky boli navrhnuté v CAD programe Fusion 360.

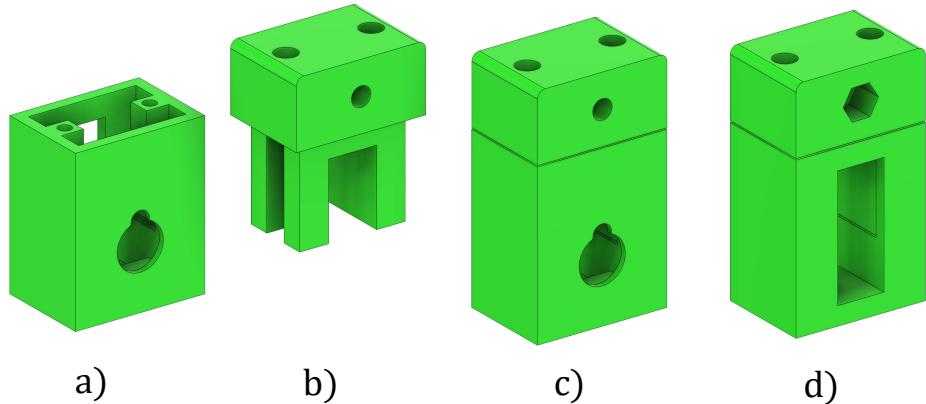


Obr. 2.15: 3D model zariadenia

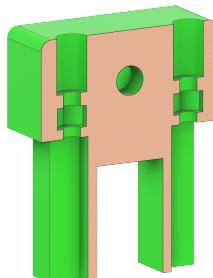
Držiak serva je prvým prvkom a nachádza sa priamo na PCB doske. Ako vyplýva z názvu, slúži na uloženie servo motora. Môžeme ho vidieť na obrázku 2.16 označený ako prvok a). Pri umiestnení servo motora v danom diely sme sa snažili aby sa poloha hriadeľa nachádzala v čo možno najnižšej polohe, aby sme tak šetrili miesto a minimalizovali rozmery zariadenia. Pripevnenie k prvku je prevedené pomocou závrtných skrutiek. Tiež sme sa rozhodli o schovanie servo motora do vnútra prvkova a jeho vkladanie cez zadnú stranu, čím jednak šetríme priestor a zároveň pri montáži nedochádza k problému s káblami servo motora, ktoré majú dostatočne veľa priestoru a nemalo by dôjsť k ich poškodeniu. Okrem toho sa v prvkove nachádza drážka, ktorá slúži na vedenie druhého dielu pri vykonávaní translačného pohybu smerom nahor a nadol, čím je splnená funkcia napínania remeňa.

Druhým prvkom - posuvný držiak prevodového kolieska (obr. 2.16, prvok b)) je už spomenutý diel, ktorý sa zasúva do prvého dielu. Translačný pohyb je zabezpečený 2 skrutkami M3 o dĺžke 20 mm, ktoré sa zapierajú o spodný diel a tým zabezpečujú zdvih celého dielu. Výška zdvihu sa nastavuje zaskrutkovávaním skrutiek do samoistiacich matíc, ktoré sú schované v samotnom telese (obr. 2.17). Ich uloženie sme vykonali zastavením tlače v danej vrstve, v ktorej sa v telese nachádzajú a po vložení matíc bola tlač znova obnovená. Aby nedochádzalo k postupnému uvoľňovaniu spoja pri prevádzke zariadenia, čo

by malo za následok uvoľnenie napäťia v remeni, rozhodli sme sa pre voľbu samoistiacich matíc. Sila, ktorou je teleso ťahané smerom dole, je vyvolaná tlakom remeňa na prevodové koliesko, ktoré je pomocou skrutky M5 uchytené na tomto prvku. V zadnej časti môžeme vidieť výrez na umiestnenie matice pre daný skrutkový spoj. Teda okrem napínania remeňa je jeho úlohou taktiež uchytanie veľkého prevodového kolieska. Na obrázku 2.16 môžeme tiež vidieť uloženie druhého dielu v prvom diele z prednej c) a zadnej d) strany.



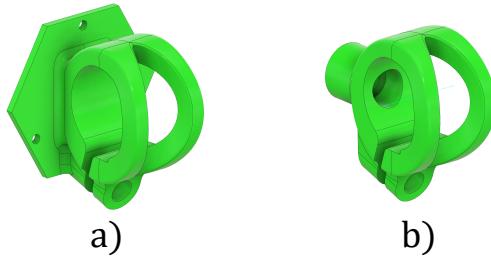
Obr. 2.16: Držiak serva a posuvný držiak prevodového kolieska



Obr. 2.17: Uloženie samoistiacich matíc

Ďalším prvkom je uchytanie trubičky na prevodové koliesko (obr. 2.18, prvok a). Ide o aktualizáciu a miernu úpravu pôvodnej verzie tohto prvku, ktorú môžeme vidieť na obrázku 2.18 ako prvok b). Keďže pôvodne sa zachytával priamo na hriadeľ servo motora, zmenou prešla práve časť uchytania prívku. Pripevnenie držiaka ku koliesku sme sa rozhodli riešiť pomocou 3 samorezných skrutiek. Pre zabezpečenie jeho presného uloženia násme tvar držiaka navrhli tak, aby pri montáži zapadol do kolieska, v ktorom sa nachádza výrez rovnakého tvaru.

Poslednými prvkami sú samotné prevodové kolieska, ktoré splňajú nami vypočítané parametre (prevodový pomer). Môžeme ich vidieť na obrázku 2.19.



Obr. 2.18: Držiak trubičky

Na koliesku o priemere 10 mm sa nachádza po celom vonkajšom obvode ozubenie dimenzované pre náš remeň a pre upevnenie kolieska k hriadeľu serva sme namodelovali vnútorné ozubenie, ktoré zodpovedá zúbkovaniu hriadeľa. Uchytanie k servo motoru po nasunutí kolieska na hriadeľ je prevedené pomocou skrutky M2. Aby sme pri jeho 3D tlači dosiahli požadovaný tvar museli sme použiť iný typ tlače ako pri ostatných prvkoch. Ide o SLA (StereoLitografia) systém, kedy sa používa fotocitlivá živica, ktorá sa vytvrdzuje laserom. Týmto spôsobom sa vytvárajú vrstvy až vznikne výsledný produkt [11]. V porovnaní s FMD (Fused Deposition Modeling) tlačou, ktorá bola použitá pri ostatných prvkoch, dosahuje väčšie presnosti a detaily – nanášaná vrstva je menšia. FMD tlač pracuje na princípe modelovania pomocou nanášania roztaveného vlákna, kedy sa hlava pohybuje v súradničiach x, y, z a tak po vrstvách postupne vytvára výsledný prvok. Koliesko o priemere 50 mm má po obvode drážku pre vedenie remeňa, ktorá bráni jeho vysunutiu. Upevnenie remeňa na koliesko je zabezpečené pomocou výrezu v tvare profilu remeňa, a nachádza sa v hornej časti kolieska. Po nasunutí remeňa do výrezov je následne ešte poistený priskrutkovaním malej plôšky k samotnému koliesku, ktorá na remeň pôsobí prítlačnou silou. Upevnenie kolieska na konštrukciu modelu je prevedené cez ložisko, aby sa mohlo koliesko voľne pohybovať pri otáčaní servo motora. Ložisko je uložené vo výreze v strede kolieska, kde sa zadnou stenou vonkajšieho krúžka opiera o koliesko a k modelu je pripojené skrutkou M5, ktorá sa zapiera o zadnú stranu jeho vnútorného krúžka.

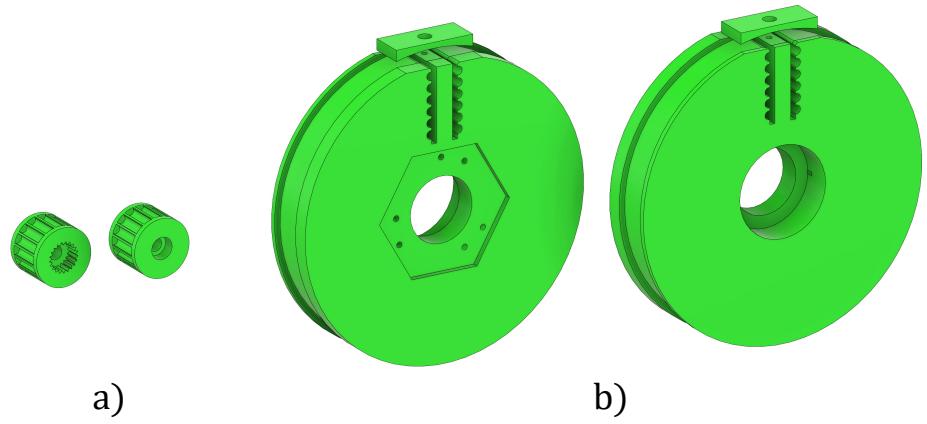
2.6 Zoznam komponentov

V tabuľke 2.5 môžeme vidieť zoznam použitých prvkov, s ich špecifikáciami a označením v schéme zapojenia. Kvôli pridaniu ďalšieho snímača – gyroskopu, a výmene servomotora, cena oproti poslednej verzii výrazne narastla a momentálne sa pohybuje okolo 30 eur na jedno zariadenie.

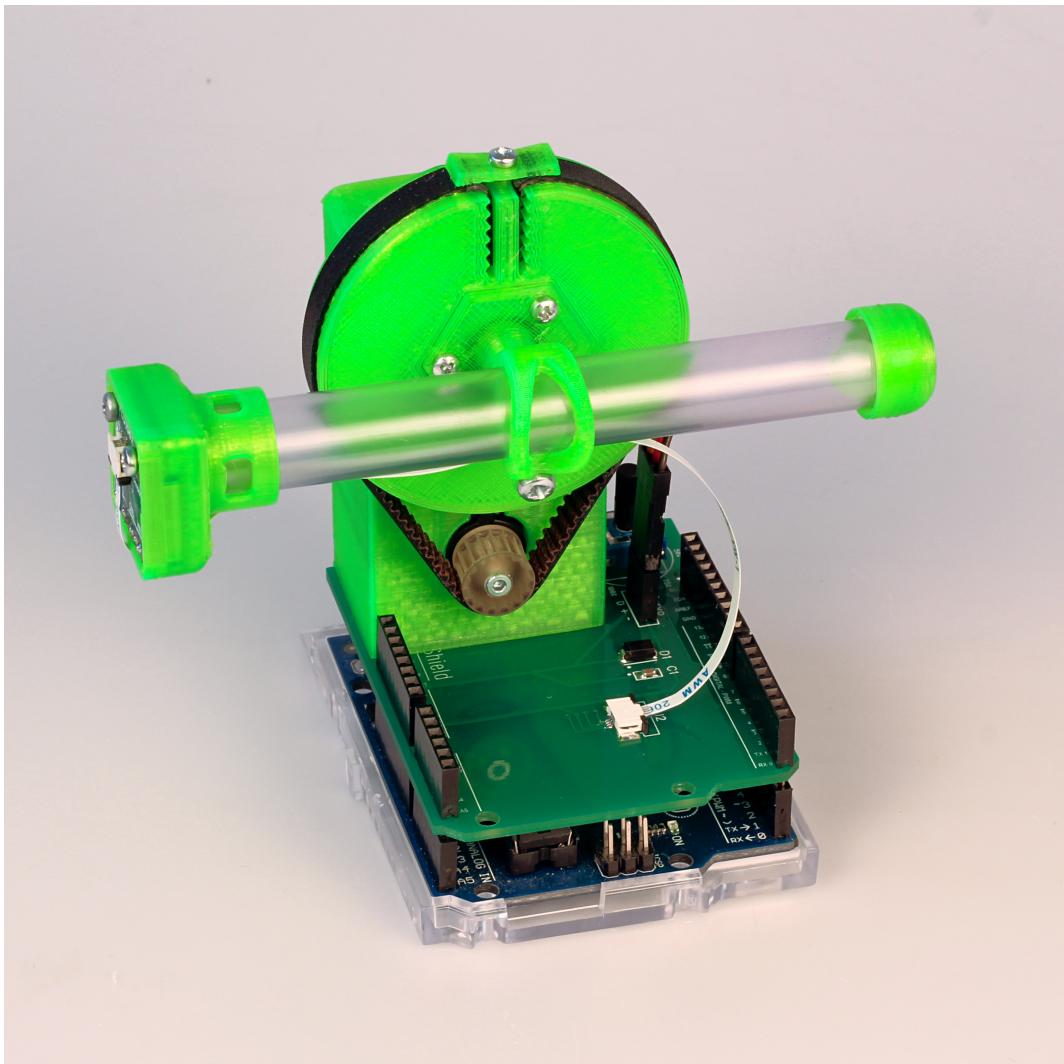
Na nasledujúcich obrázkoch 2.20 sa nachádza už zostavený hardvér. Celkové rozmery sa v porovnaní so staršími verziami zväčšili, na čom má najväčší podiel hlavne použitie prevodu rotačného pohybu. Tiež treba spomenúť, že nejde o úplne kompletnú verziu BOBShield R3, z dôvodu nedodania objednávky komponentov. Cýbajú ale len menšie zmeny na PCB doskách. Na obrázku teda môžeme vidieť hardvér R3, zostavený na doske verzie R2.

značka	názov	špecifikácia	počet
(a)	Servomotor	SAVOX - SH, Digitálne	1
(e)	ToF snímač	VL6180X, laserový snímač vzdialenosťi	1
(f)	Gyroskop	MPU6050,	1
(c)	Potenciometer	10 kOHM, 250mW, CA9MV 10K ACP	1
(b)	Regulátor napäťia	LDO, ST732M28R, 2.8V	1
(g),(k)	Kondenzátor	100 uF, 6.3V, 0805	2
(h)	Kondenzátor	0,1 uF, 6.3 V, 0805	1
(i)	Kondenzátor	0,47 uF, 6.3 V, 0805	1
(l)	Kondenzátor	4,7 uf, 6.3 V, 0603	1
(m),(n)	Rezistor	10kOHM, 100 V, 0805	2
-	Gulička	PP (polypropylén), 8 mm	1
(p)	Kábel	FFC kábel, 0.5 mm, 4 kontakty	1
(q)	Ložisko	priemer 5 mm	1
-	PCB doska	2- vrstvová, FR4, 1.6 mm	1
(j)	Dióda	DO214AC	1
-	Skrutka	Din 7984 M3 x 20	2
-	Skrutka	Din 7984 M5 x 35	1
-	Skrutka	Din 912 M2 x 6	1
-	Samorezná skrutka	Din 7981C 2,2 x 6,6	3
-	Samorezná skrutka	Din 7981F 2,9 x 6,5	2
-	Matica	Din 985 M3	2
-	Matica	Din 934 M5	1
-	Podložka	Din 125A PA M3	2

Tabuľka 2.5: Zoznam použitých komponentov



Obr. 2.19: Prevodové kolieska



Obr. 2.20: BOBShield R3

3 BoBshield API

3.1 API pre Arduino IDE

Arduino IDE je voľne dostupný softvér vyvinutý pre zjednodušenie programovania Arduino dosiek, či už ide o písanie kódu alebo jeho nahrávanie na dosku. Pri poslednej verzii sa API pre Arduino IDE skladalo z knižnice BOBShield.h a príkladov na identifikáciu systému alebo PID riadenia systému. Nasledujúca podkapitola hovorí o úpravách knižnice, ktoré sme implementovali, prečo bolo potrebná dané zmeny aplikovať a aký majú dopad na chod zariadenia. Tiež sa venuje analýze filtrácie vhodnej pre naše zariadenie, ktorá má za úlohu vyhľadiť krivku meraných hodnôt na výstupe zo systému.

3.1.1 Filtrácia

Pri meraniach presnosti ToF snímača pre rôzne materiály, povrchy a farby guľičiek sme mohli pozorovať šum, ktorý nám pri nameraných hodnotách vznikal. Hoci tento nežiadúci jav nie je možné celkom odstrániť, môžeme jeho dopad na namerané hodnoty výrazne znížiť pomocou implementácie jednoduchej filtrácie nameraných hodnôt vhodnou metódou. Pre našu aplikáciu sme testovali 3 typy filtrácie a to pomocou aritmetického priemeru, aritmetického plávajúceho a váženého plávajúceho priemeru. Pri testovaní sme postupovali nasledovne: Základom bolo získanie súboru nameraných hodnôt, na ktorý budeme môcť aplikovať filtre a následne navzájom porovnať ich výstupy. Vstup je pre všetky typy filtrov identický a preto môžeme spoľahlivejšie rozhodnúť, ktorý z filtrov si zvolíme. Tieto hodnoty vstupu sme získali meraním polohy guľôčky pri jej posune z jednej koncovej polohy trubičky k druhej, pri minimálnom naklonení trubičky. Hodnoty boli získané a merané ešte na pôvodnej verzii zariadenia – BOBShield R2.

Pri aritmetickom priemere sme z určitého počtu nameraných hodnôt urobili aritmetický priemer a výstupom bola 1 hodnota, z čoho vyplýva, že frekvencia, s ktorou sme merali je násobne väčšia ako frekvencia, získavania filtrovaných hodnôt. To môže mať za následok spomalenie systému z hľadiska rýchlosťi reakcie na zmenu polohy guľôčky. Vlastnosti daného typu filtráciu vieme ovplyvniť počtom nameraných hodnôt, z ktorých robíme aritmetický priemer. Zvyšovanie počtu hodnôt má za následok väčšie vyhľadenie výslednej krivky, no zároveň spôsobuje určité oneskorenie, čo môže byť pozorované hlavne v prípade keď guľôčka rýchlo mení svoju polohu.

Rozdiel pri plávajúcim aritmetickom priemere je ten, že je založený na princípe zberu a ukladaní určitého počtu hodnôt, pričom pri každom ďalšom meraní sa nám najstaršia hodnota vymaže a pozíciu prvej hodnoty zaujme najnovšie nameraná hodnota. Z tohto súboru hodnôt sa nám pri každom meraní vypočíta aritmetický priemer, teda frekvencia merania a frekvencia získavania nameraných hodnôt je identická.

Vážený plávajúci priemer je založený na princípe je založený na rovnakom princípe práce s nameranými hodnotami ako plávajúci aritmetický priemer. Rozdielom je, že každá hodnota má inú váhu (ako vyplýva z názvu), teda má rozdielny vplyv na výslednú hodnotu. Výsledná hodnota sa následne vypočíta podľa nasledujúceho vzorca [13]:

$$priemer = \frac{\sum_{i=1}^N x_i \cdot p(x_i)}{\sum_{i=1}^N p(x_i)} = \frac{x_1 \cdot p(x_1) + x_2 \cdot p(x_2) + \cdots + x_N \cdot p(x_N)}{p(x_1) + p(x_2) + \cdots + p(x_N)} \quad (3.1)$$

Vlastnosti filtrácie teda vieme ovplyvňovať jednak počtom hodnôt, z ktorých vážený priemer robíme, ale aj zmenou parametrov ovplyvňujúcich váhu jednotlivých hodnôt, teda vplyv na výsledok.

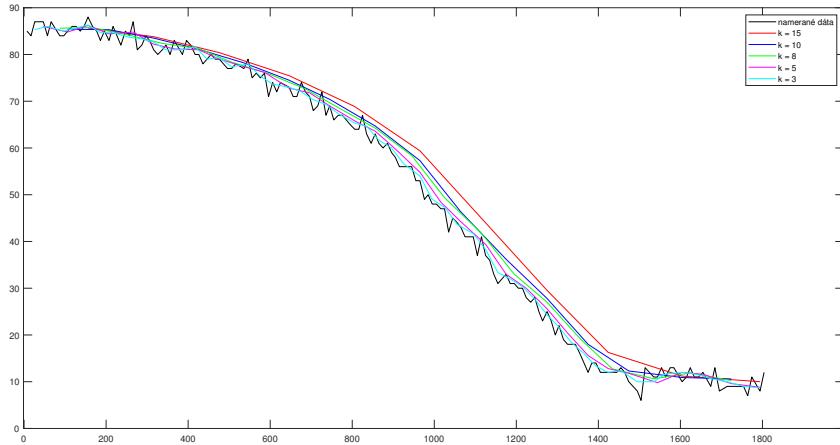
Vzorku hodnôt, na ktorej sme porovnávali vplyv jednotlivých typov filtrov, sme si vykreslili do grafu v prostredí MATLAB spolu s grafmi hodnôt získaných po aplikácii filtrov. Následne sme menili parametre filtrov, či už išlo o počet hodnôt, z ktorých sme tvorili priemer alebo pri váženom priemere alebo o nastavenie parametrov vázenia.

Pri procese hľadania parametrov pre vážený priemer sme uvažovali faktor zabúdania. Ide o bezrozmerné číslo z intervalu 0 až 1 a od jeho výberu závisí aké budú parametre jednotlivých váh pre dané hodnoty. Ak si zvolíme faktor zabúdania 0.9, väčšiu váhu budú mať práve staršie dátá. Naopak ak si zvolíme číslo blížiace sa k 0, dôraz bude kladený na najnovšie hodnoty. Jednotlivé koeficienty váh sa na základe faktora zabúdania získajú zo vzorca 3.2, kde λ predstavuje faktor zabúdania. Pre rýchle systémy je lepšou voľbou výber menších hodnôt faktora [6].

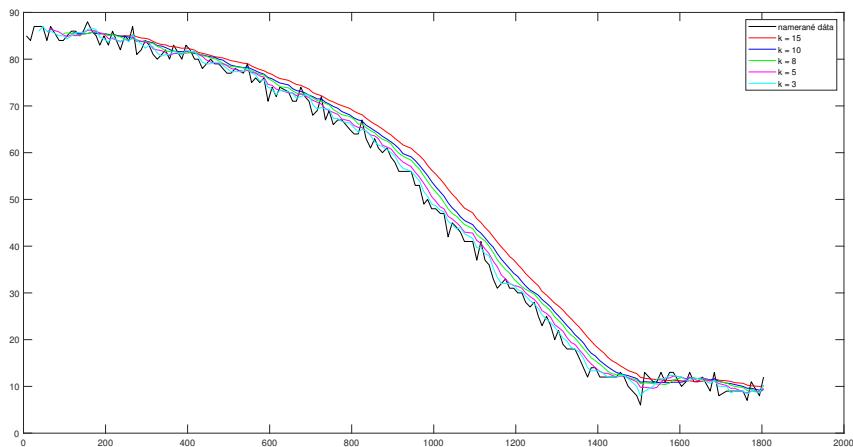
$$p_{n+1} = p_n \cdot \lambda + 1 \quad (3.2)$$

Na obrázku 3.1 môžeme vidieť graf filtrácie pomocou aritmetického priemera pre rozdielny počet hodnôt, z ktorých sme priemer získali. Hoci vyhľadenie krivky vyzerá veľmi dobre, oproti plávajúcemu priemeru, hodnoty získavame s oveľa nižšou frekvenciou, čo vytvára oneskorenie reakcie na rýchlu zmenu polohy guľôčky. Preto je určite vhodnejšou variantou jeden z typov plávajúcich priemerov.

Na ďalšom grafe (obr. 3.2) môžeme vidieť filtráciu prostredníctvom plávajúceho aritmetického priemera. Rovnako sme testovali rôzne počty hodnôt, z ktorých sme daný priemer robili. Z grafu môžeme vidieť, že pri vyšších počtoch hodnôt sa krivka výrazne lepšie vyhľadí, no zároveň je na grafe očividné



Obr. 3.1: Aritmetický priemer

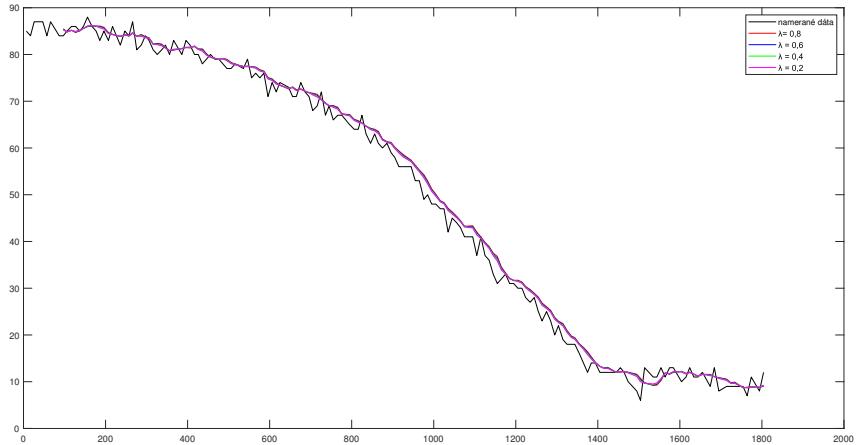


Obr. 3.2: Plávajúci aritmetický priemer

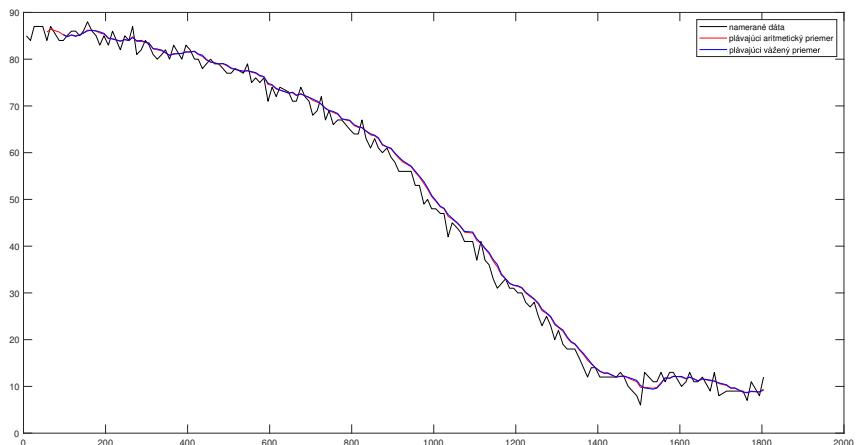
oneskorenie oproti nefiltrovaným dátam. Z daných možností, ktoré sme testovali, sme sa rozhodli pre filtráciu pomocou 5 posledných hodnôt. Krivka dobre kopíruje nefiltrované dáta z hľadiska oneskorenia a zároveň dosahuje prijateľné vyhladenie šumu.

Na grafe (obr. 3.3) vidíme porovnanie filtrácie pomocou plávajúceho váženého priemeru z 5 hodnôt pre rozdielne faktory zabúdania. Môžeme vidieť, že rozdiel medzi jednotlivými krivkami je minimálny. Rozhodli sme sa zvoliť faktor zabúdania $\lambda = 0.4$, keďže potrebujeme rýchlu reakciu na zmenu aktuálnej vzdialenosťi, čiže dôraz na posledné hodnoty.

Na obrázku 3.4 môžeme vidieť porovnanie filtrácie plávajúceho aritmetického a váženého priemeru spolu s pôvodnými, nefiltrovanými nameranými hodnotami. Môžeme vidieť, že rozdiel medzi krivkami je minimálny. Jednoduchšou voľbou z hľadiska výpočtov je určite aritmetický priemer, čím budeme šetriť



Obr. 3.3: Plávajúci vážený priemer



Obr. 3.4: Porovnanie plávajúceho aritmetického a váženého priemeru

pamäť mikropočítača a výsledok bude takmer identický. Z tohto dôvodu sme sa rozhodli pre filtračiu pomocou plávajúceho aritmetického priemera, počítaného z 5 posledných nameraných hodnôt.

Po výbere vhodného filtra pre odstránenie šumu sme v knižnici BOBS-hield.h vytvorili novú funkciu – filterSensorRead(). Jej úlohou je teda odstránenie šumu pri čítaní hodnôt zo snímača a tým zlepšenie kvality riadenia systému.

3.1.2 Knižnica

Zmeny prevedené na hardvéri sa samozrejme odrážajú aj v softvérovej časti zariadenia. Z tohto dôvodu je potrebné upraviť už existujúcu knižnicu BOBShield.h, vytvorenú pre jednoduchú prácu so zariadením, aby bola kompatibilná aj s na-

šou verziou zariadenia.

Prvou vecou, ktorú sme urobili, bola tvorba premennej `BOB_SHIELD_VERSION` definovanej na začiatku knižnice. Slúži pre nastavenie verzie zariadenia, ktorú práve používame. Momentálne v nej môžu byť nastavené 2 hodnoty a to: 2 pre verziu R2 a 3 pre verziu R3. Túto premennú využívame pri funkciách, ktorých parametre sa pre jednotlivé verzie zariadenia líšia, ako je napríklad `calibration` alebo `actuatorWrite`. Jedná sa teda hlavne o funkcie, v ktorých sa nachádza nastavenie rozsahu pohybu servomotora.

Pri funkcii `actuatorWrite` bolo potrebné zmeniť rozsah pohybu servomotora a tiež prevod hodnôt zadávaných do funkcie na hodnoty, ktoré budú vstupovať do servomotora pri jeho riadení. Pri pôvodnej verzii išlo a jednoduchý prevod za použitia funkcie `map`, kedy sa vstupné hodnoty z rozsahu -30 až 30 menili na hodnoty 65 až 125, pričom išlo a prevod celých čísel na celé čísla.

Kedže naším cieľom bolo zlepšiť presnosť otáčania trubičky a teda zvýšiť počet hodnôt, ktoré môže nadobúdať, rozhodli sme sa pre implementáciu prevodu, v ktorom 1° na servomotore sa rovná $0,2^\circ$ na trubičke. Z tohto dôvodu nie je možné využiť funkciu `map`, kedže už ide o prevod desatinných čísel na celé. Pre prevod sme sa rozhodli použiť funkciu `mapFloat` – naprogramovanú v knižnici `AutomationShield`, ktorá prevádzka hodnoty desatinných čísel opäť na desatinné čísla. Teda z premennej `degree`, ktorá slúži na posielanie hodnôt na servomotor, sa nám stane desatinné číslo. Servomotor však dokáže pracovať len s celými číslami a preto je potrebné použiť funkciu `round()`, ktorá nám premenňuje `degree` zaokrúhlí na celé číslo, ktoré už následne môžeme použiť pri riadení servomotora. Za hodnoty vstupu sme si zvolili rozsah -17 až 17 a za výstupný rozsah uhlov na servomotore 10 až 170.

Funkcia `calibration` funguje na princípe hľadania krajných hodnôt vzdialenosí guľôčky v trubičke pri jej pohybe. Najprv hľadá minimálnu hodnotu. Nastaví trubičku do krajnej hodnoty rozsahu jej otáčania a zmeria 100 hodnôt, z ktorých následne urobí aritmetický priemer a ten vloží do premennej `minCalibrated`. To isté sa deje aj pri hľadaní maximálnej hodnoty, ktorú vloží do premennej `maxCalibrated`. Jediné čím sa funkcia líši vo verzii R3 od pôvodnej verzии sú krajné hodnoty otočenia trubičky. Tie bolo teda potrebné upraviť pre naš hardvér.

3.2 API pre Simulink

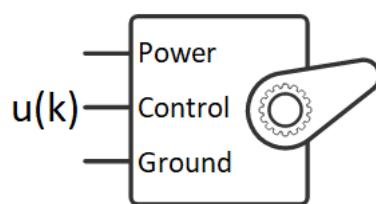
Simulink je grafické programovacie prostredie založené na MATLABe a určené pre modelovanie, simuláciu a analýzu dynamických systémov. Jeho hlavným nástrojom je rozhranie pre vytváranie grafických blokových diagramov a sada blokových knižník, určených pre rôzne typy aplikácií. Ide o nástroj často využívaný pri automatickom riadení. Využíva sa aj pri digitálnom spracovaní signálov. Táto časť práce sa venuje API vytvorenému v tomto prostredí a určenému pre riadenie nášho zariadenia.

3.2.1 Knižnica

Pre tvorbu rozhrania pre programovanie nášho zariadenie v prostredí Simulink je potrebné vytvoriť vlastnú knižnicu, ktorá obsahuje potrebné funkcie pre komunikáciu so zariadením. Táto sa svojou formou líši od knižnice vytvorenej pre prostredie Arduino IDE. V porovnaní s ňou neobsahuje kód ako taký ale skladá sa z jednotlivých blokov, ktorých úlohou je zastrešiť určitú funkciu. Následným spájaním blokov do diagramu vytvárame výsledný program, ktorý je pri komplikácii nahraný do mikropočítača. Knižnica je teda klasický súbor pre Simulink model (.slx) obsahujúci dané bloky a do zoznamu knižníc v programovacom prostredí je pridaný prostredníctvo MATLAB kód. Aby sme knižnicu vedeli dostúpiť je potrebné spustiť súbor s názvom InstallMatlabAnd-Simulink nachádzajúci sa v knižnici AutomationShield. Jeho úlohou je nastaviť všetky cesty tak, aby program vedel nájsť potrebné súbory pre implementáciu knižnice. Knižnica BOBShield pre Simulink už bola rozpracovaná. V našej práci sme ju sfunkčnili a doplnili o potrebné funkcie. Následne sme ju vložili medzi ostatné knižnice nachádzajúce sa AutomationShield API pre Simulink.

Jednotlivé bloky nachádzajúce sa v BOBShield knižnici teda slúžia rovnako ako funkcie naprogramované v knižnici pre Arduino IDE a zastrešujú rovnaké úlohy.

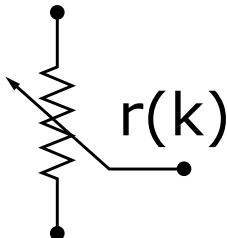
Prvým blokom je **Actuator Write** 3.5, ktorého úlohou je riadenie servomotora. Spracúva vstupnú hodnotu vchádzajúcu do bloku a následne nastavuje uhol otočenia servomotora v rozsahu, ktorý sme preň určili. Pri nastavovaní parametrov bloku si môžeme zvoliť typ vstupu vchádzajúceho do bloku a to z 2 možností – v percentách alebo vzdialenosťi reprezentovanej v mm. Dôležitým parametrom je taktiež rozsah uhlov, ktorý pre blok nastavíme. Môžeme buď nechať defaultne prednastavené hodnoty v rozsahu -30° až $+30^\circ$ alebo zvoliť možnosť manuálneho nastavenia vlastného rozsahu, kde si zvolíme minimálny a maximálny uhol otočenia.



Obr. 3.5: Blok Actuator Write

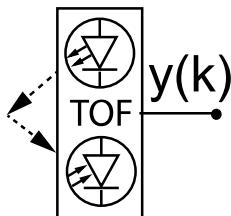
Blok **Reference Read** 3.6 nám v diagrame reprezentuje potenciometer umiestnený na BOBShielde. Slúži na získavanie hodnôt z potenciometra, ktorý môže byť využitý ako referenčná hodnota pri riadení systému. Po kliknutí pravým tlačidlom myši na daný blok v schéme si môže užívateľ nastaviť spôsob akým budú hodnoty z potenciometra prezentované v Simulinku. Je možné zvoliť si z 3 možností, ktoré môžeme vidieť v sekcií Readout type. Prvou je získanie hodnoty v percentoch, kde hodnoty, ktoré získavame, budú z intervalu od 0 po 100 v závislosti od polohy bežca potenciometra. Ďalšou možnosťou je získa-

vanie hodnôt v podobe napäťia v rozsahu od 0 po 5 V. Poslednou dostupnou možnosťou nastavenia čítania hodnôt z potenciometra je analógová hodnota z intervalu od 0 po 1023. V závislosti od potreby užívateľa blok poskytuje teda určitú flexibilitu v závislosti od aplikácie, v ktorej ho chceme využiť.



Obr. 3.6: Blok Reference Read

Ďalším blokom je **Sensor Read** 3.7 a slúži nám na získavanie aktuálnej polohy guľôčky v trubičke. Ide o blok slúžiaci na komunikáciu mikropočítača s naším ToF snímačom VL6180X prostredníctvom I2C protokolu. Používateľ si môže vybrať z typov výstupných hodnôt zo snímača. Prvou možnosťou je vzdialenosť v mm, ktorá nám dáva vzdialenosť guľôčky od snímača v trubičke. Ďalšou z možnosti je pozícia guľôčky v mm. Ide o úpravu vzdialenosťi, kde sa za počiatočnú pozíciu považuje koncový bod trubičky, v ktorom sa nenačádza ToF snímač, teda výstupné hodnoty sú inverzné voči vzdialenosťi. Pri voľbe výstupu v percentách sú potrebné kalibračné hodnoty aby vedel blok učiť minimálnu a maximálnu meranú hodnotu pre daný typ telesa v trubičke. Tie je možné zadať nižšie v okne parametrov bloku. Môžeme si zvoliť buď defaultne nastavené hodnoty – 0 až 100 mm alebo nastaviť nami namerané krajné hodnoty pomocou manuálneho zadávania.

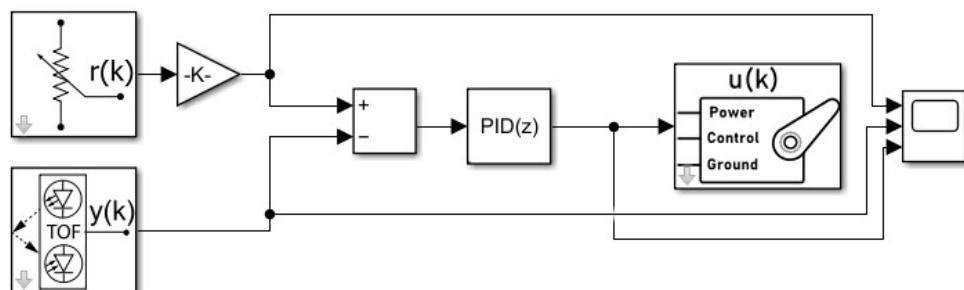


Obr. 3.7: Blok Sensor Read

Posledným blokom je blok **Filtered Sensor Read**, ktorého úlohou je vyhľadenie šumu snímača pri meraní vzdialenosťi guľôčky. Ide o implementáciu filtrácie signálu, ktorú sme si zvolili na základe testovaní viacerých typov filtrácií na nami nameranej zložke dát. Na vyhľadenie šumu používame plávajúci aritmetický priemer počítaný z posledných piatich nameraných hodnôt. Daný blok sme vytvorili ako subsystém, do ktorého sme vložili blok Sensor Read a výstup z neho sme vložili do bloku Moving Average, v ktorom sme si za veľkosť okna (parameter window length) zvolili 5 hodnôt a jeho výstup sme zaviedli na výstup zo subsystému.

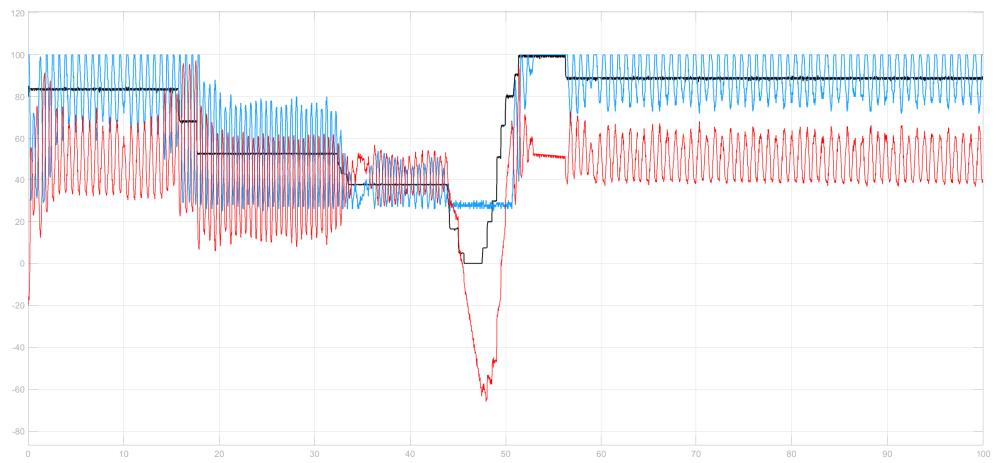
3.2.2 PID príklad

Riadenie systému pomocou PID regulátora je jeden z najznámejších typov riadenia systému. Naše zariadenie sme sa teda rozhodli riadiť pomocou PID regulátora v uzavretej riadiacej slučke (obr.3.8). V princípe sa v schéme nachádza slučka so spätnou väzbou zo zariadenia, ktorá porovnáva aktuálny stav systému s referenčnou hodnotou a na základe toho vyhodnocuje akčný zásah do systému. V našom zapojení referenčnú hodnotu dostávame z bloku Reference Read, ktorý nám ponúka aktívne ju meniť pomocou otáčania bežcom na potenciometri. Spätnú väzbu o stave zariadenia sme získavame prostredníctvom bloku Filtered Sensor Read, ktorý nám hovorí o aktuálnej pozícii guľôčky v zariadení. Blok PID nám následne na základe vypočítanej chyby určí akčný zásah do systému a pošle ho do bloku Actuator Write, ktorý následne nastaví pozíciu servomotora do požadovanej hodnoty.



Obr. 3.8: Schéma zapojenia - príklad PID riadenia

Na obrázku 3.9 môžeme vidieť výstup z priebehu riadenia, kedy krivka čiernej farby reprezentuje referenčnú hodnotu, modrá aktuálnu pozíciu guľôčky a červená akčný zásah do systému. Môžeme vidieť, že zariadenie na zmenu referenčnej hodnoty aktívne reaguje a snaží sa pomocou zásahu do systému regulovať polohu guľôčky v danej polohe. Guľôčka zatiaľ len osciluje okolo referenčnej polohy ale po dôkladnom naladení by mal byť systém schopný udržať jej pozíciu priamo na požadovanej hladine. Momentálne z hľadiska problémov pri komunikácii hardvéru s rozhraním Simulink sme neboli schopní aplikovať riadny a detailný proces ladenia PID regulátora. No môžeme vidieť, že daný proces PID riadenia pomocou programu Simulink je možné uskutočniť a knižnica je plne funkčná.



Obr. 3.9: Výstup z priebehu riadenia

4 Záver

Cieľom tejto bakalárskej práce bolo zrevidovať existujúci hardvér, navrhnúť a realizovať jeho zmeny, vytvoriť API v programovacom prostredí Simulink a zrealizovať príklady riadenia systému na novom hardvéri a v novom programovacom prostredí.

Podarilo sa nám výrazne vylepšiť hardvérovú časť zariadenia, kde sme pomocou aplikácie prevodu medzi servomotorom a trubičkou zvýšili presnosť jej otáčania z 1° na $0,2^\circ$. Tiež sme dôkladnou analýzou výberu vhodnej guličky pre náš systém znížili smerodajnú odchýlku pri meraní snímača. Z hľadiska softvéru sme pridali do pôvodnej knižnice v Arduino IDE funkcie na filtrovanie šumu a upravili sme pôvodné funkcie aby boli kompatibilné s obidvoma verziami zariadenia R2 a R3. Vytvorili sme API pre Simulink a overili jeho funkčnosť na príklade PID riadenia systému.

Taktiež sme sa snažili o tvorbu API v programovacom prostredí MATLAB, ku ktorému sme vytvorili knižnicu, no narazili sme na problém v komunikácii medzi programovacím prostredím MATLAB a ToF snímačom, ktorý sme v našom zariadení použili.

Do budúcnosti zariadenie stále ponúka priestor na zlepšenie. Môže dôjsť k tvorbe nových príkladov LQ riadenia pre obe API, taktiež k procesu identifikácie systému na novom hardvéri. Zaujímavé by bolo aj porovnanie funkčnosti starého servomotora s novým prevodom oproti novému servu. Taktiež môže dojsť ku tvorbe funkcie na nastavenie vodorovnej polohy trubičky pri kalibrácii systému, ktorá nemohla byť vytvorená z dôvodu nedodanie komponentov na jej realizáciu.

Literatúra

- [1] CTMS - Control tutorials for MATLAB and SIMULINK. Ball and beam: System modeling. Online., 2021. 12.5.2022,<https://github.com/gergelytakacs/AutomationShield/wiki/BoBShield>.
- [2] DFROBOT. 9g metal gear micro servo (1,8kg). Online., 2021. 22.5.2022,<https://www.dfrobot.com/product-1338.html>.
- [3] G. Takacs. Bobshield. Online., 2021. 22.4.2022,<https://github.com/gergelytakacs/AutomationShield/wiki/BoBShield>.
- [4] InvenSense Inc. Mpu-6000 and mpu-6050 product specification revision 3.4. Online., 2013. 21.4.2022,<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [5] ISL Products International Ltd. Servo motor fundamentals. Online., 2021. 20.5.2022,<https://islproducts.com/design-note/servo-motor-fundamentals/>.
- [6] MathWorks. Sliding window method and exponential weighting method. Online., 2020. 3.7.2014, <https://www.mathworks.com/help/dsp/ug/sliding-window-method-and-exponential-weighting-method.html>.
- [7] MOUSER Electronics. Mpu-6050. Online., 2021. 22.5.2022,<https://www.mouser.sk/ProductDetail/TDK-InvenSense/MPU-6050?qs=u4fy%2FsgLU9014B5JgyQFvg%3D%3D>.
- [8] SAVOX-SERVO. Savox - servo - sh-0253 - digital - dc - motor. Online., 2021. 22.5.2022,<https://www.savox-servo.com/Servos-c-1338/Brushed-Motor-c-1340/Savox-Servo-SH-0253-Digital-DC-Motor/>.
- [9] STMicroelectronics. Vl6180x - proximity and ambient light sensing module. Online., 2016. 21.4.2022,<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [10] STMicroelectronics. St732 datasheet. Online., 2021. 28.4.2022,<https://www.mouser.sk/ProductDetail/STMicroelectronics/ST732M28R?qs=MLItCLRbWsxtXiiWQqtIMw%3D%3D>.
- [11] Technické vybavenie Libre. Typy 3d tlače. Online., 2020. 26.5.2022, <https://www.hwlibre.com/sk/Typy-3D-tla%C4%8De/>.

- [12] A. Vargová. *BoBShield: a miniature "ball on beam" experiment*. PhD thesis, Slovenská technická univerzita v Bratislave, Strojnícka fakulta, Bratislava, Slovenská republika, April 2021.
- [13] Wikipedia. Weighted arithmetic mean. Online., 2022. 26.5.2022, https://en.wikipedia.org/wiki/Weighted_arithmetic_mean.